

**Elektronická sbírka řešených
příkladů s interaktivní 3D grafikou k
výuce Diferenciálního počtu funkcí
více proměnných**

**Electronic collection of solved
problems with examples of
interactive 3D graphics for teaching
the differential calculus of functions
of several variables**

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 6. května 2009

.....

Ráda bych na tomto místě poděkovala všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla, zvláště pak mé vedoucí bakalářské práce RNDr. Petře Šarmanové, Ph.D. .

Abstrakt

Cílem této bakalářské práce je vytvořit webovou galerii příkladů s interaktivní 3D grafikou vytvořenou za pomoci systému Maple a programu JavaView pro podporu výuky diferenciálního počtu funkcí dvou proměnných, která ilustruje základní teoretické pojmy od vrstevnic přes tečnou rovinu až po Taylorův polynom. Takto vytvořená a okomentovaná grafika přispívá k pochopení nových pojmů a k objasnění geometrického významu řešených úloh. Základem galerie jsou interaktivní grafické objekty vytvořené v Maplu, vyexportované pomocí knihovny JavaViewLib a následně prezentovány na webu programem JavaView, jako Java applety.

Klíčová slova: Maple, JVL, JavaView, HTML stránky, jsMath, export grafů

Abstract

The main goal of the Bachelor thesis is to create the web gallery collection of examples with interactive 3D graphic objects created using the Maple software and computer programme JavaView to support the education of differential calculation of function of two variables which illustrates the basic theoretical conceptions from level curve, tangential plane to Taylor polynomial. The created and commented graphic makes it easier to understand new conceptions and also to make the outcome of geometric problem solution clear. The gallery consists of interactive graphic objects created by Maple, exported using the JavaViewLib library and presented on web using the computer programme JavaView as Java Applets.

Keywords: Maple, JVL, JavaView, HTML pages, jsMath, graphs export

Seznam použitých zkratk a symbolů

CSS	– Cascading Style Sheets
JVL	– Java View Library
PHP	– Hypertext Preprocessor
HTML	– Hyper Text Markup Language
XML	– Extensible Markup Language

Obsah

1	Úvod	6
2	Maple	7
2.1	Instalace	7
2.2	Zápisník	9
3	Mapleovská knihovna JavaViewLib	13
3.1	Náčtení knihovny JVL	13
3.2	Export grafů	14
4	Java View	25
4.1	Ovládání appletů	25
4.2	Control panel	26
5	HTML	28
5.1	PHP	28
5.2	Začlenění interaktivní grafiky do HTML	29
5.3	Animace	32
5.4	JsMath	34
6	Ukázky příkladů	40
6.1	Rozvržení webové galerie	40
6.2	Vrstevnice a řezy	42
6.3	Parciální derivace	44
6.4	Směrové derivace	47
6.5	Tečná rovina	48
6.6	Taylorův polynom	49
7	Závěr	50
8	Reference	52

Seznam tabulek

1	Maple příkazy	11
2	Options	12
3	Export část 1.	14
4	Export část 2.	15
5	Export a zobrazení grafu	15
6	Zkratky pro ovládání appletů	25
7	Control Panel	27
8	Parametry appletu část 2.	30
9	Parametry appletu část 1.	31
10	Parametry animace	33
11	Easy load	36
12	Parciální derivace	49

Seznam obrázků

1	Worksheet File Association Selector	8
2	Worksheet	10
3	3D objekt v Maple	10
4	Maple graf	17
5	JavaView graf	17
6	Export grafu s osou z.	17
7	Maple Graf 2D	18
8	JavaView Graf 2D	18
9	Export 2D grafu s osou z.	18
10	Vrstevnice v Maple	19
11	Vrstevnice v JavaView	19
12	style=contour Maple	19
13	style=contour JavaView	19
14	style=patchcontour Maple	19
15	style=patchcontour JavaView	19
16	style=wireframe Maple	20
17	style=wireframe JavaView	20
18	style=hidden Maple	20
19	style=hidden JavaView	20
20	style=point Maple	20
21	style=point JavaView	20
22	Graf.jvx	21
23	Zobrazení grafu Graf v Maple	21
24	Graf.mpl	22
25	Nastavení Transparency	22
26	Úprava tečny a bodu.	23
27	Nastavení os grafu.	23
28	Control panel.	26
29	Rozvržení HTML stránky	28
30	flow.1.jvx	32
31	flow.2.jvx	32
32	flow.3.jvx	32
33	flow.1.jvx	33
34	flow.2.jvx	33
35	flow.3.jvx	33
36	jsMath pomocí tagu DIV a SPAN	35
37	jsMath-button	37
38	jsMath-dialog	37
39	jsMath-message	37
40	jsMath-button	39
41	jsMath-dialog	39
42	jsMath-message	39

43	Náhled úvodní stránky kapitoly Vrstevnice a řezy	40
44	Řez rovinou $z = 4$	42
45	Znázornění vrstevnic v rovině xy	42
46	Řez rovinou $y = 0$	42
47	Znázornění řezu v rovině xz	42
48	Řez rovinou $x = 0$	43
49	Znázornění řezu v rovině yz	43
50	Parciální derivace podle x v bodě A	44
51	Řez funkce rovinou $y = 1$ a tečna $z = 2x$	44
52	Parciální derivace funkce podle y v bodě B	45
53	Řez rovinou $x = 1$ a tečna $z = -2y$	45
54	Směrová derivace podle vektoru u v bodě A	47
55	Řez rovinou, rovnoběžnou s osou z a procházející přímkou $A + tu$, kde $t \in R$	47
56	Funkce a tečná rovina v bodě $A = (0, 0)$	48
57	Funkce a tečná rovina v bodě $A = (\frac{1}{2}, \frac{1}{2})$	48
58	Taylorův polynom 1. stupně.	49
59	Taylorův polynom 2. stupně.	49

Seznam výpisů zdrojového kódu

1	Export formátu htm.	16
2	Ukázka členění webové stránky pomocí PHP.	29
3	Začlenění interaktivní grafiky do HTML.	29
4	Animace	32
5	Sázení matematiky pomocí tagu DIV a SPAN.	35
6	Sázení matematiky pomocí dolarů.	35
7	Atributy jsMath	37

1 Úvod

V dnešní době existuje mnoho knih, skript a publikací podporujících výuku nejen matematiky. Naproti tomu se stává pro mnohé studenty samozřejmostí vyhledávání informací na internetu. Vezmeme-li pro srovnání tištěné publikace a elektronické texty mají nesporné výhody. Jeden text slouží zároveň neomezenému počtu studentů. Studenti tak ušetří čas vyhledáváním v knihovnách a čekáním až bude kniha k dispozici. Elektronické texty jsou snadno šířitelné a vyhledavatelné. Dochází také k jistému propojení informací a vzniku návazností textů. Elektronické publikace jsou snadněji aktualizovatelné a odpadají tak náklady na výtisk nových vydání. Výhod elektronických textů bychom našli spousty, ale největší zbraní těchto aplikací jsou dynamické prvky, jako jsou Java applety nebo animace vytvořené programem Flash. Tím jsou 2D obrázky v podobě tištěného textu nahrazeny 3D objekty, s nimiž je možné nadále pracovat. Tyto elektronické texty mají široké uplatnění v oborech, které vyžadují od studentů představivost. Mezi něž patří mimo jiné matematika. Zařazení dynamických prvků je vhodné zejména pro ta odvětví matematiky, která se zabývají 3D objekty, ať už z geometrického či analytického hlediska.

V této bakalářské práci si ukážeme jednu z možností vytvoření elektronického textu. Cílem bude vytvoření webové galerie týkající se problematiky diferenciálního počtu funkcí více proměnných, která ilustruje základní teoretické pojmy vrstevnic, parciální derivace, směrové derivace, tečnou rovinu a Taylorův polynom, pomocí interaktivní 3D grafiky. S programy Maple a JavaView bylo dosaženo požadovaných výsledků. Vyexportované grafy jsou umístěné na web prostřednictvím Java appletu. Tento postup nevyžaduje znalost jazyka Java pro tvůrce a také klade minimální nároky na uživatele. Jediným požadavkem je webový prohlížeč s běhovým prostředím Java, ale dnes je toto rozhraní implementováno téměř do každého webového prohlížeče.

Náplní bakalářské práce je popsání vytvoření webové galerie se všemi jejími prvky. Text bakalářské práce je členěn do šesti kapitol. Po úvodní kapitole následuje kapitola, která se zabývá popisem programu Maple a jeho možnostmi vytváření grafů. Třetí kapitola vysvětluje propojení programu Maple s programem JavaView pomocí mapleovské knihovny JavaViewLib. Zde je uvedena nejen instalace a popis funkcí této knihovny, ale i možné problémy, vzniklé při práci s ní. Běhovým prostředím Java View pro zobrazování interaktivní grafiky se zabývá další kapitola. Pátá kapitola popisuje tvorbu webových stránek pomocí programovacího skriptu PHP, začlenění interaktivní grafiky na tyto stránky a tvorbu matematického textu pomocí skriptu jsMath. Skript jsMath převádí latexovský kód matematického zápisu do výstupu. Tento způsob tak nahrazuje ruční převod matematických symbolů do obrázků. Nakonec jsou uvedeny ukázky příkladů, které jsou dostupné na webu. Stránky jsou k dispozici na adrese

<http://am.vsb.cz/vondrova> nebo na CD přiloženém k této práci.

2 Maple

Systém Maple je jedním ze systémů počítačové algebry (Computer Algebra Systém). Umožňuje provádět jak symbolické a numerické výpočty, tak vytvářet grafy funkcí, programovat vlastní funkce či procedury, ukládat data v několika formátech (např. LaTeX, HTML, MATHML, ...) a dokonce provádět export do programovacích jazyků (např. C, Fortran 77, ...).

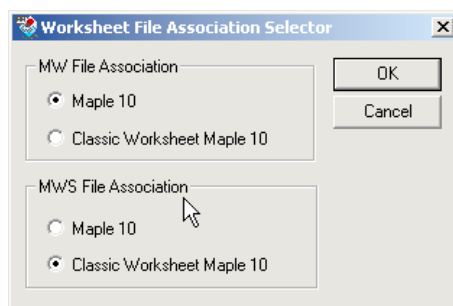
Funkce implementované v Maplu pokrývají širokou oblast matematiky od základů lineární algebry, diferenciálního a integrálního počtu, přes diferenciální rovnice, geometrii až k logice. Je možné také vykreslovat grafy funkcí jedné (dvou) proměnných, funkcí určených parametrickými rovnicemi. Vizualizace funkce v Maplu je dána desítkami předdefinovanými grafickými funkcemi s proměnlivým počtem parametrů.

2.1 Instalace

Systém Maple je distribuován v různých verzích i různých typech instalací. Následující návod je pro instalaci Maple 10. Instalace je popsána obecně, proto je použitelná i pro jiné verze.

1. Spustíte instalaci, buď se spustí automaticky z CD, nebo z menu položky **Instaluj Maple 10**. Nespustí-li se instalace, přejděte k souboru
`...\Windows\Disk1\InstData\VM\instalMaple10.exe`.
2. Zobrazí se okno s instrukcemi **Introduction**, nic nenastavujeme a na další položku přejdeme tlačítkem "next".
3. Výběr instalačního adresáře **Choose Install Folder**. Instalujeme Maple 10 do adresáře, který neobsahuje mezery např. `C:\Maple10` a přejdeme dále tlačítkem "next".
4. V **Client or Server Settings** zvolte "yes" pokud budete instalovat licenční utility, v druhém případě ponechejte "no".
5. **License Manager Server** zde uvedeme licenční server, na kterém je dostupná licence k tomuto programu.
6. V položce **Profile Settings** zvolte Single-user Profile.
7. **Install Desktop Shortcuts** - "yes" - v případě, že chcete mít ikonu na ploše.
8. **Maple Worksheet Association** - "no" (doporučuji zvolit pro reasociaci typu souborů).
9. **Watcom Installation** - "yes" pro instalaci C kompilery a určete umístění kompilery např. `C:\Maple10\watcom-1.3`.
10. **Pre-Installation Summary** pokračujte "next".

11. Vynechali-li jste krok 4. a 5. zobrazí se vám další položka **Maplesoft Activation** - krok aktivace znamená zadání tzv. purchase codu, který jste obdrželi s koupí Maple. Aktivace vyžaduje připojení k internetu a je velmi jednoduchá, stačí jen sledovat pokyny instalátoru.
12. Po kompletní instalaci resetujte počítač.
13. Spusťte Maple 10 a systém by se měl bez problémů rozběhnout - v systému Windows XP (při aktivním firewallu) je potřeba povolit přístup k serveru maple (měli byste být operačním systémem k odblokování vyzváni).
14. Změňte registraci typu souborů v nabídce **Start** → **Maple 10** → **tools**. Vyberte nástroj Worksheet File Association Selector a objeví se dialogové okno, kde zvolte:
 - MW association - Maple 10
 - MWS association - Classic Worksheet Maple 10



Obrázek 1: Worksheet File Association Selector

Při instalaci na operační systém Windows Vista dochází k jistým potížím, jelikož tato verze není ani určena pro tento systém. Tento nedostatek je odstraněn s novou verzí Maple 12. Pokud nemáme dostupnou tuto verzi a potřebujeme Maple 10 nainstalovat na operační systém Windows Vista, odstraníme potíže nastavením kompatibility:

- Po instalaci klikneme pravým tlačítkem na ikonu programu Maple 10 nebo Classic Worksheet Maple 10.
- Vyberte příkaz nabídky Vlastnosti a klepněte na kartu Kompatibilita.
- Nastavte režim kompatibility a v rolovacím menu vyberte Windows XP.

Po správné instalaci se nám zobrazí tři aplikace:

- Maple 10
- Classic Worksheet Maple 10
- Maple Calculator

Maple Calculator je klasická kalkulačka rozšířená o spoustu funkcí, založena na standardním rozhraní GUI. Hlavní aplikací jsou dva programy Maple 10 a Classic Worksheet Maple 10.

Dříve byla k dispozici pouze jedna aplikace a to Classic Worksheet Maple, která je založena na standardním grafickém rozhraní GUI. Toto grafické rozhraní bylo velmi dobře použitelné a přehledné, ale bohužel vyžadovalo poměrně hlubokou znalost programovacího jazyka. Postupem času a vývojem stále nových verzí Maple, byl tento nedostatek odstraněn doplněním o další aplikaci Maple 10. Nyní plně záleží na uživateli, která aplikace mu vyhovuje více. Obě dvě obsahují stejné funkce a spolupracují.

Já jsem ke své práci zvolila aplikaci zápisníku Classic Worksheet Maple. Příkazy se zde píšou pomocí příkazové řádky, což mi osobně vyhovuje nejvíce. Tím nejdůležitějším důvodem mého výběru byla časová náročnost na zpracování příkazů v Maple 10, proto dále budeme pracovat v zápisníku Classic Worksheet Maple 10. Následné převedení příkazů do aplikace Maple 10 je bezproblémové.

2.2 Zápisník

Jak už jsem uvedla výše, pro práci jsem si zvolila Classic Worksheet Maple 10. Ikona tohoto zápisníku, pokud jste nastavili při instalaci, se nachází na ploše nebo v nabídce **Start** → **Maple 10** → **Classic Worksheet Maple 10**. Zde se nachází také už zmiňovaný Maple Calculator a Maple 10. V této kapitole se budeme zabývat pouze popisem zápisníku.

2.2.1 Základy práce

Po spuštění programu se otevře v okně prázdný soubor, který je nazýván zápisníkem (worksheet). Objeví se nám okno programu, kde vidíme titulek, nabídkovou lištu a nástrojovou lištu.

V nabídce View je položka Palletes a v ní nalezneme různé palety, které nám urychlí přístup k některým funkcím.

Nový dokument začíná vždy znakem prompt:

```
> |
```

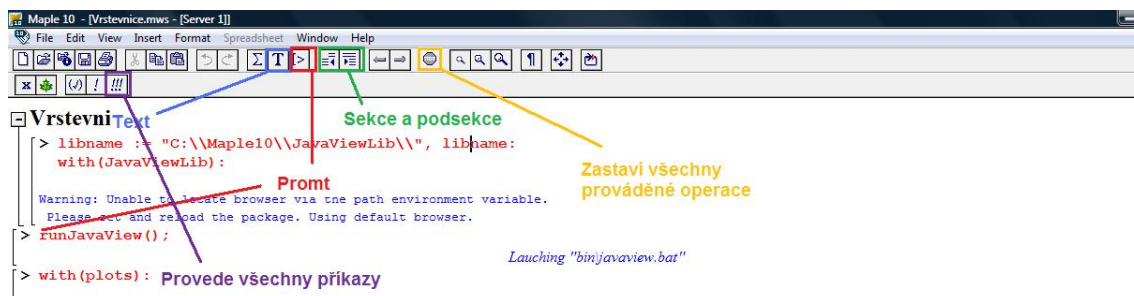
Za tento znak píšeme mapleovské příkazy, které chceme zpracovat a ukončíme středníkem. Toto ukončení je nutné, protože jinak Maple očekává pokračování předchozího příkazu. Pro příklad si spustíme nápovědu:

```
> help(packages);
```

Stiskneme ENTER a příkaz bude vykonán, tj. program vypíše provedenou operaci na další řádek a kurzor se přesune za následující prompt. Pokud chceme následné vypsání potlačit, ukončíme příkaz místo středníkem dvojtečkou. Pro přerušování zpracování příkazu stiskneme tlačítko STOP na liště nástrojů, popř. klávesu break.

Pokud chceme na další řádek pokračovat s příkazem a vynechat prompt, použijeme kombinaci kláves SHIFT+ENTER.

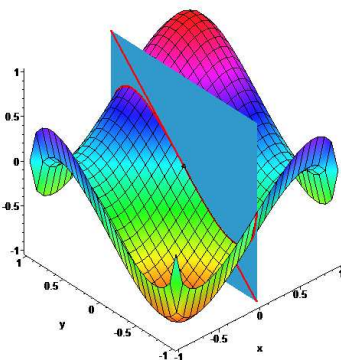
Maple je case sensitive, tedy rozlišuje malá a velká písmena. Pracujeme-li ve Windows, v názvu cesty zpětné lomítko nahradíme dvojtečkou.



Obrázek 2: Worksheet

Uvedeme si příklad vytvoření 3D objektu:

```
> f:=x*sqrt(1-x*x)+y*sqrt(1-y*y):
> f1:=plot3d(f, x=-1..1, y=-1..1, axes=framed, color=f):
> bod:=pointplot3d([0,0,0], symbol=circle, color=black):
> krivka:=spacecurve([0, t, t*sqrt(1-t*t)], t=-1..1,
axes=NORMAL, thickness=3, color=red):
> rovina:=plot3d([0,y,z], y=-1..1, z=-1..1,
axes=framed, color=COLOR(RED, 0.1960, 0.6000, 0.8000)):
> tecna:=spacecurve([0, t, t], t=-1..1, axes=NORMAL,
thickness=3, color=RED):
> display(f1,bod,rovina,krivka,tecna);
```



Obrázek 3: 3D objekt v Maple

Použité mapleovské příkazy jsou vysvětleny v následující tabulce, doplněny o další základní příkazy:

2D objekty	
Zápis	Popis
<code>contourplot($f(x, y)$, $x = a..b$, $y = c..d$, options)</code>	Zobrazí vrstevnice funkce $f(x, y)$ v rovině xy , kde $x \in (a, b)$ a $y \in (c, d)$.
<code>plot(f, $x = x_0..x_1$, options)</code>	Zobrazí graf funkce proměnné x , kde $x \in (x_0, x_1)$.
<code>pointplot(L, options)</code>	Zobrazí bod v rovině. Za parametr L dosazujeme zobrazený bod např. $[0, 0]$. Zapsat můžeme i více bodů najednou $\{[0, 0], [1, 2] \dots\}$.
<code>textplot(L, options)</code>	Zobrazí text v prostoru. Za parametr L píšeme souřadnice, kde má být text umístěn a samotný text jako třetí souřadnici např. $[0, 0, 'Text']$. Zapsat můžeme i více textu najednou $\{[0, 0, 'Text1'], [1, 2, 'Text2'] \dots\}$.
<code>display(L, insequence=true, options)</code>	Zobrazí více objektů najednou, nebo animaci, zvolíme-li <code>insequence=true</code> . Za proměnnou L dosazujeme názvy 2D objektů např.: $\{A, B\}$.
3D objekty	
Zápis	Popis
<code>contourplot3d($f(x, y)$, $x = a..b$, $y = c..d$, options)</code>	Zobrazí vrstevnice funkce $f(x, y)$ v prostoru, kde $x \in (a, b)$ a $y \in (c, d)$.
<code>plot3d($f(x, y)$, $x = a..b$, $y = c..d$, options)</code>	Zobrazí graf funkce v prostoru, kde $x \in (a, b)$ a $y \in (c, d)$.
<code>SpaceCurve(C, $t = a..b$, options)</code>	Zobrazí křivku v prostoru, kde C je parametrické vyjádření křivky $f(t)$ např. $[1 + t, 1 + t, 2 + 2t]$, kde $t \in (a, b)$.
<code>pointplot3d(L, options)</code>	Zobrazí bod v prostoru. Za parametr L píšeme zobrazovaný bod např. $[0, 0, 0]$. Zapsat můžeme i více bodů najednou $\{[0, 0, 0], [1, 1, 2] \dots\}$.
<code>textplot3d(L, options)</code>	Zobrazí text v prostoru. Za parametr L píšeme souřadnice, kde má být text umístěn a samotný text jako čtvrtou souřadnici např. $[0, 0, 0, 'Text']$.
<code>display(L, insequence=true, options)</code>	Zobrazí více objektů najednou, nebo animaci, zvolíme-li <code>insequence=true</code> . Za proměnnou L dosazujeme názvy 3d objektů např.: $\{A, B\}$.

Tabulka 1: Maple příkazy

Parametr options, který máme uveden u všech předchozích příkazů, umožňuje nastavení plno dalších parametrů objektu (barva, osy, styl...). Ty základní si uvedeme v následující tabulce:

Zápis	popis
axes=f	Nastavuje osy, za f dosazujeme boxed, normal, frame, nebo none. Výchozí je none.
color=c nebo colour=c	Nastavuje barvu objektu, za c dosazujeme barvu buď slovně (color=blue, red ...), číselně pro RGB model (color=COLOR(RGB, .5607, .7372, .5607)) nebo předpisem jako funkci (color= x^2) ¹ .
style=s	Nastaví, jak se má objekt vykreslit (drátěný model, bodově ...), za s dosazujeme contour, hidden, line, patch, patchcontour, patchnograd, point, nebo wireframe. Výchozí je patch.
thickness=n	Nastaví tloušťku křivky, za n dosazujeme kladné číslo. Výchozí je 0.

Tabulka 2: Options

1) Barva se vykresluje v hladinách vrstevnic funkce. Podle hladin barva se mění a využije se celé barevné spektrum.

3 Mapleovská knihovna JavaViewLib

Jedná se o knihovnu pro Maple 6 a vyšší, zveřejňovanou pod zkratkou JVL. JavaViewLib umožňuje export interaktivní 3D grafiky do formátu MPL (interní datový formát Maplu) a JVX (oficiální formát programu JavaView, založený na jazyku XML). Kromě těchto výstupů je umožněn přímý export do HTML kódu a okamžité zobrazení interaktivní 3D grafiky pomocí webového prohlížeče.

JVL se tak stává spojovacím článkem mezi systémem Maple a programem pro manipulaci s 3D grafickými objekty JavaView. Máme-li správně nainstalovány programy Maple a JavaView (instalace Maplu byla popsána v předchozí kapitole a o instalaci programu JavaView budeme mluvit v kapitole následující), pak můžeme začít využívat funkce této knihovny.

Knihovna se dá stáhnout zdarma z adresy <http://www.javaview.de/> v sekci download. Balíček si tedy stáhneme a rozbalíme do adresáře, kde jej pro další práci budeme chtít mít. Autor doporučuje umístit do instalačního adresáře Maple. Předpokládejme tedy umístění:

```
C:\Maple10\JavaViewLib
```

Pro první práci s JVL si spustíme ukázkovou stránku JVLExport.htm uloženou v adresáři htm. V našem případě tedy: `file:///C:/Maple10/JavaViewLib/htm/sample.htm`.

3.1 Načtení knihovny JVL

Dříve než začneme využívat funkce této knihovny, je třeba k ní nastavit cestu. Uložíme si cestu ke knihovně JVL do proměnné libname:

```
> libname:=" C:\Maple10\JavaViewLib",libname:
```

Po té knihovnu načteme:

```
> with(JavaViewLib);
```

Abychom tyto příkazy nemuseli zadávat při každém spuštění Maplu, můžeme přepsat konfigurační soubor maple.ini, který bude obsahovat výše uvedené příkazy. Ty se spustí automaticky při každém startu Maplu. Instalační balík JVL obsahuje tento soubor maple.ini v podadresáři bin. Stačí jej přepsat v textovém editoru. Přidáme i příkaz pro načtení mapleovské knihovny plots, která obsahuje funkce pro zobrazování grafů. Tento soubor přesuneme do adresáře Maplu bin.win nebo LIB. Obsah souboru maple.ini:

```
libname:="C:\\Maple10\\JavaViewLib",libname:
with(JavaViewLib):
with(plots)
```

Po načtení knihovny můžeme prohlížeč JavaVlew přímo v maple spustit příkazem:

```
> runJavaView();
```

Do závorek se dále uvozují 3D nebo 2D objekty vytvořené v Maple viz kapitola 2.

3.2 Export grafů

Prvním krokem je převedení mapleovských grafů do formátu, se kterým umí JavaView pracovat. Jak už jsme si řekli, jednou z úloh JavaViewLib je tyto dvě aplikace propojit.

Maple používá k popisu generované grafiky vlastní datovou strukturu, užívající jen prostého textu. První z možností exportu pomocí JavaViewLib využívá přímo tohoto zápisu, téměř nezměněný ho přepíše do vnějšího souboru s příponou MPL. Další možností propojení je převést datovou strukturu Maplu do formátu JVX, což je mateřský formát JavaView, založený na značkovacím jazyce XML.

K exportu mapleovských grafů využijeme příkazy knihovny JavaViewLib skupiny export, které převádí grafy do vybraného formátu, a příkazy skupiny run, které po exportu navíc graf zobrazí. Tyto příkazy máme popsány v následující tabulce:

Zápis	Popis
exportMPL(graf): exportJVX(graf): exportHTM(graf): runJavaView(graf):	Graf JVLEExport.mpl se uloží v pracovním adresáři knihovny JVL. Pokud jste postupovali při instalaci přesně podle návodu zde: C:\Maple10\JavaViewLib\mpl\JVLEExport.mpl obdobně jako mpl (jvx\JVLEExport.jvx). obdobně jako mpl (htm\JVLEExport.htm). obdobně jako mpl (mpl\JVLEExport.mpl) a navíc se hned po exportu graf zobrazí.
exportMPL(graf, "graf"): exportJVX(graf, "graf"): exportHTM(graf, "graf"): runJavaView(graf, "graf"):	Graf graf.mpl se uloží v pracovním adresáři knihovny JVL. Pokud jste postupovali při instalaci přesně podle návodu zde: C:\Maple10\JavaViewLib\mpl\graf.mpl obdobně jako mpl (jvx\graf.jvx). obdobně jako mpl (htm\graf.htm). obdobně jako mpl (mpl\graf.mpl) a navíc se hned po exportu zobrazí. ²
exportMPL(graf, "C:\\nazev\\"): exportJVX(graf, "C:\\nazev\\"): exportHTM(graf, "C:\\nazev\\"): runJavaView(graf, "C:\\nazev\\"):	Graf se uloží zde C:\\nazev\JVLEExport.mpl. obdobně jako mpl (... \JVLEExport.jvx). obdobně jako mpl (... \JVLEExport.htm). obdobně jako mpl (... \JVLEExport.mpl) a navíc se hned po exportu graf zobrazí.

Tabulka 3: Export část 1.

2) Použijeme-li příponu .jvx vyexportuje se nám graf v JVX formátu. Obdobně i v dalších příkazech runJavaView.

Zápis	Popis
<code>exportMPL(graf, "C:\\nazev\\graf"):</code> <code>exportJvX(graf, "C:\\nazev\\graf"):</code> <code>exportHTM(graf, "C:\\nazev\\graf"):</code> <code>runJavaView(graf, "C:\\nazev\\graf"):</code>	Uloží se soubor zde <code>C:\\nazev\\graf.mpl</code> . obdobně jako <code>mpl (...\\graf.jvx)</code> . obdobně jako <code>mpl (...\\graf.htm)</code> . obdobně jako <code>mpl (...\\graf.mpl)</code> a navíc se hned po exportu graf zobrazí.
<code>exportHTMLite</code>	Používáme stejné příkazy jako u <code>exportHTM</code> . V HTML stránce se použije odlehčená verze appletu <code>JavaView</code> a to <code>javLite.jar</code> .

Tabulka 4: Export část 2.

Knihovna JvL nabízí ještě možnost po exportu spustit rovnou webový prohlížeč. K tomu je potřeba nastavit cestu ke spustitelnému souboru prohlížeče. Tuto informaci mu předá systémová proměnná `PATH`, do které je třeba tuto cestu přidat. Používáme-li Internet Explorer, nastavíme cestu k spustitelnému souboru `explorer.exe`, který je většinou umístěn v adresáři `C:\\ProgramFiles\\InternetExplorer` do proměnné `PATH` přes **Ovládací panely** → **Systém** → **Upřesnit** → **Proměnné prostředí**.

Obdobně můžeme nastavit i jiný prohlížeč, podporované jsou Fire Fox, Netscape a Opera. Ještě nám chybí nastavit výchozí prohlížeč v maple příkazem:

```
> runJavaView(set(BR="IE");
```

Pro Fire Fox IE změníme za FF, pro Netscape za NS a pro Operu za OP. A nyní nám zbývají příkazy pro export:

Zápis	Popis
<code>runApplet:</code>	Používáme stejné příkazy jako u <code>runJavaView</code> . Po exportu se spustí nastavený webový prohlížeč s appletem <code>JavaView</code> .
<code>runApplet:</code>	Používáme stejné příkazy jako u <code>runJavaView</code> . V HTML stránce se použije odlehčená verze appletu <code>JavaView</code> a to <code>javLite.jar</code> .

Tabulka 5: Export a zobrazení grafu

Máme nyní tři možnosti exportu (MPL, JVX, HTM). U formátu HTM se jedná o přímý export do HTML kódu. Chceme-li vytvořit rychle jednoduchou HTML stránku bez jakýchkoliv úprav, využijme tento formát. Ukázka HTML kódu, který byl vytvořen automaticky při použití příkazů:

```
> bod:=pointplot3d([0,0,0],symbol=box):
> exportHTM(bod);
```

Záměrně jsem zvolila graf o nejmenším počtu bodů, tedy přímo pouze jeden bod, aby kód nebyl příliš rozsáhlý.

```

<HTML>
<HEAD>
<TITLE> – JVL</TITLE>
</HEAD>
<BODY BGCOLOR='#FFFFFF'>
<P ALIGN='CENTER'>
<TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 ALIGN='CENTER'>
  <TR><TD>
<APPLET CODE='javaview.class' ARCHIVE='jars/javaview.jar,jars/jvx.jar' CODEBASE='..'
  WIDTH='400' HEIGHT='400' ID='JVLAPPLET' ALT='JVL – MAPLE Export'>
    <PARAM NAME='DepthCue' VALUE='hide'>
    <PARAM NAME='Background' VALUE='255_255_255'>
    <PARAM NAME='Border' VALUE='hide'>
    <PARAM NAME='Maple' VALUE='PLOT3D(POINTS([0.,0.,0.]),SYMBOL(BOX))'>
  </APPLET>
</TD><TD>
</TD><TR><TD>
</TD><TD>
</TD>
</TABLE>
</P></BODY></HTML>

```

Výpis 1: Export formátu htm.

Jak vidíme v kódu, applet je vkládán do tabulky html tagem <TABLE> pro lepší začlenění do HTML kódu. Do tagu <PARAM NAME='Maple' VALUE=''> je uložen nezměněný obsah formátu MPL, tudíž i vlastnosti zobrazovaného objektu jsou totožné. Vytváříme-li rozsáhlé webové aplikace, doporučuji exportovat objekty v maple do samostatných souborů JVX nebo MPL a vkládat podle návodu v kapitole 5. Oba formáty si popíšeme blíže v následujících kapitolách.

3.2.1 Formát MPL

Vyhovuje-li nám vykreslení grafu v Maple a nepotřebujeme měnit vzhled a další parametry, vystačíme si s formátem MPL. Bohužel se některé funkce po exportu JavaView nezobrazí, které si popíšeme podrobněji. Občas jsem se setkala také s případem, že graf nešel vyexportovat vůbec. Důvodem byl většinou nesprávný název proměnné, do které byl graf uložen anebo špatně zvolen rozsah os. Značnou výhodou MPL oproti formátu JVX je poměrně malá velikost. Podívejme se blíže na problémové případy, se kterými jsem se setkala. Podrobněji se problémem při exportech zabývá webová galerie na stránkách <http://www.javaview.de/maple/validate/index.htm>, která ukazuje srovnání zobrazení v JavaView a původní rendering Maple.

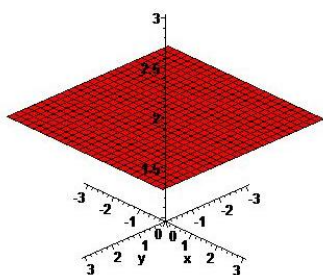
3.2.1.1 Vykreslení os grafu Chceme-li exportovat graf s osami do formátu mpl, mění se exportem rozsah os. Respektuje pouze nastavení mapleovského příkazu axes, jenž je popsán v tabulce 2. JavaView vždy všechny osy zkrátí na minimální délku. To může být někdy výhodou, ale není tomu tak v následující ukázce, kde si vykreslíme graf roviny $z=2$:

```
> graf:=plot3d(2, x=-3..3,y=-3..3, axes=framed, color=red):
> graf;
```

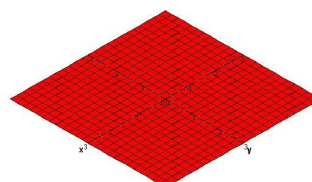
Výsledek vykreslení v Maple je zobrazen na obrázku 23. Dále si graf vyexportujeme a zobrazíme v JavaView příkazem:

```
> runJavaView(graf);
```

výsledek vidíme na obrázku 5.



Obrázek 4: Maple graf

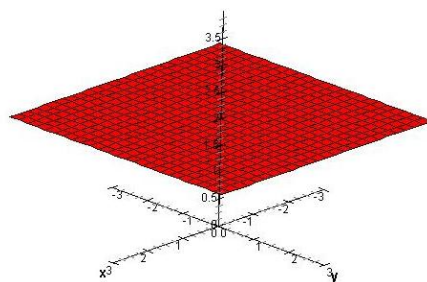


Obrázek 5: JavaView graf

Po porovnání vykreslení v obou aplikacích vidíme, že se nám po exportu osa z úplně ztratila a místo grafu $z = 2$ máme vykreslen graf $z = 0$. Nejedná se pouze o problém při vykreslování osy z , problém nastane i u ostatních os. Tento nedostatek jsem vyřešila vytvořením vlastní osy za pomoci maplevského příkazu `spacecurve` viz tabulka 1. V předchozím příkladě pro graf $z = 2$ by vykreslení osy z vypadalo takto:

```
> osa:=spacecurve([0,0,t],t=0..4, color=black):
> graf:=plot3d(2, x=-3..3,y=-3..3, axes=framed, color=red):
> Graf:=display(osa,graf):
> runJavaView(Graf);
```

Výsledek vykreslení v JavaView je zobrazen na obrázku 9. Tentýž problém nastal

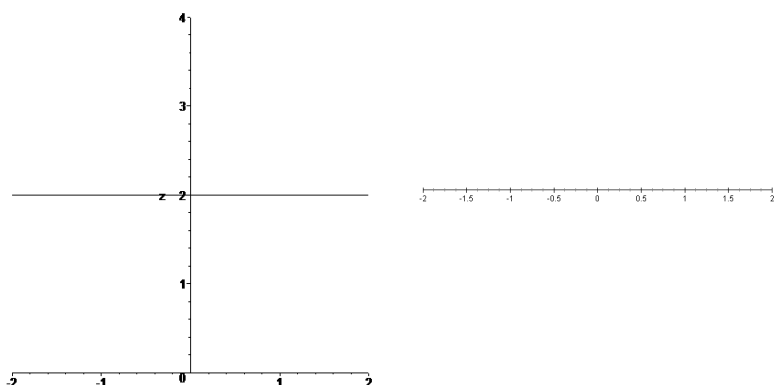


Obrázek 6: Export grafu s osou z .

také při vykreslování 2D grafů. Ukázku si uvedeme v následujícím příkladě, kde máme vykreslenou přímku o rovnici $x=2$.

```
> primka:=plot(2, x=-2..2, z=0..4, color=blue):
> primka;
> runJavaView(primka);
```

Předchozí příkazy nám provedly vykreslení grafu, jak v prostředí Maple výsledek viz Obrázek 7, tak v aplikaci JavaView viz Obrázek 8.

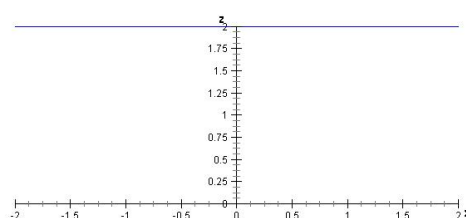


Obrázek 7: Maple Graf 2D Obrázek 8: JavaView Graf 2D

Opět snadným řešením je vytvořit si osu vlastní:

```
> osa:=plot(x=0, x=-2..2, z=0..4, color=black):
> primka:=plot(2, x=-2..2, z=0..4, color=blue):
> Graf:=display(primka,osa):
> runJavaView(Graf);
```

Výsledek exportu zobrazeného v JavaView je zobrazen na následujícím obrázku:



Obrázek 9: Export 2D grafu s osou z.

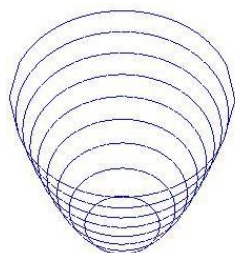
3.2.1.2 Vykreslení vrstevnic Maple nám nabízí na vykreslení vrstevnic grafu v prostoru příkaz `contourplot3d`, bohužel toto zobrazení JavaView nepodporuje. Vykreslení vrstevnic grafu v rovině příkazem `contourplot` pracuje bez problémů, tento příkaz popsán blíže v tabulce 1. Nyní si uvedeme příklad na použití mapleovského příkazu `contourplot3d`:

```
> vrstevnice:=contourplot3d(x^2+y^2, x=-3..3,
y=-sqrt(9-x*x)..sqrt(9-x*x), color=blue):
> vrstevnice;
```

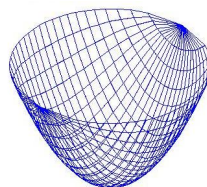
Pro export a zobrazení v JavaView použijeme příkaz:

```
> runJavaView(vrstevnice);
```

Na následujících obrázcích máme zobrazen výsledek zobrazení vrstevnic, jak v prostředí Maple viz Obrázek 10, tak v aplikaci JavaView viz Obrázek 11, kde vidíme, že aplikace JavaView zobrazení vrstevnic nepodporuje a převede na klasický drátěný model.



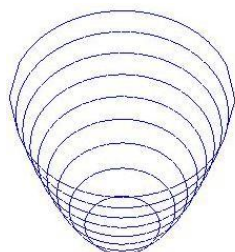
Obrázek 10: Vrstevnice v Maple



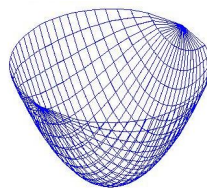
Obrázek 11: Vrstevnice v JavaView

3.2.1.3 Vykreslení stylu Maple nabízí 7 stylů vykreslování grafů, které jsou uvedeny v tabulce 2. JavaView plně podporuje pouze styly patch (nastaven implicitně) a patchnograd. Styly contour a patchcontour vykreslující na plochu vrstevnice nepodporuje JavaView vůbec, ale je možnost vykreslit vrstevnice ručním zadáváním křivek, což je velice zdlouhavé a pracné. Poslední tři styly wireframe, hidden a point jsou podporovány pouze částečně. Nyní si pro srovnání ukážeme všech 5 problémových stylů vykreslených v Maple a zároveň v JavaView:

```
> plot3d(x^2+y^2,x=-3..3,y=-sqrt(9-x*x)..sqrt(9-x*x),color=blue,
style=contour);
```

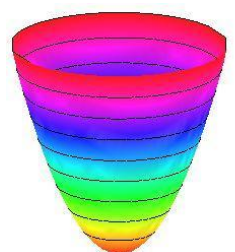


Obrázek 12: style=contour Maple

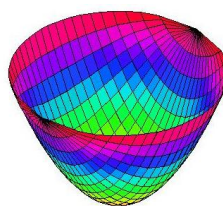


Obrázek 13: style=contour JavaView

```
> plot3d(x^2+y^2,x=-3..3,y=-sqrt(9-x*x)..sqrt(9-x*x),color=x^2+y^2,
style=patchcontour);
```

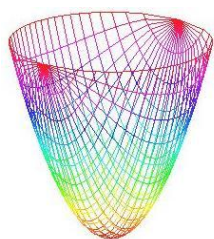


Obrázek 14: style=patchcontour Maple

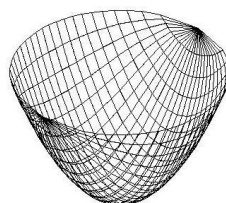


Obrázek 15: style=patchcontour JavaView

```
> plot3d(x^2+y^2,x=-3..3,y=-sqrt(9-x*x)..sqrt(9-x*x),color=x^2+y^2,
style=wireframe);
```

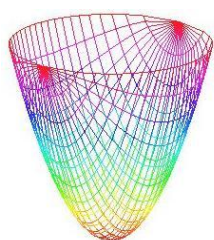



Obrázek 16: style=wireframe Maple

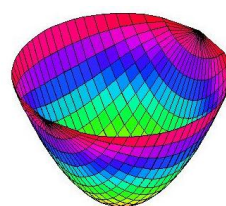


Obrázek 17: style=wireframe JavaView

```
> plot3d(x^2+y^2,x=-3..3,y=-sqrt(9-x*x)..sqrt(9-x*x),color=x^2+y^2,
style=hidden);
```



Obrázek 18: style=hidden Maple

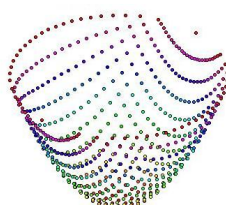


Obrázek 19: style=hidden JavaView

```
> plot3d(x^2+y^2,x=-3..3,y=-sqrt(9-x*x)..sqrt(9-x*x),color=x^2+y^2,
style=point);
```



Obrázek 20: style=point Maple



Obrázek 21: style=point JavaView

Všimněme si dalšího rozdílu, který vznikl na všech předchozích ukázkách, grafy zobrazené v JavaView mají jiný poměr os oproti grafům zobrazeným v Maple. Tento rozdíl jsme si popsali již v kapitole 3.2.1.1, chcete-li toto zobrazení změnit, postupujte podle návodu v následující kapitole 3.2.2 převodem do formátu JVX.

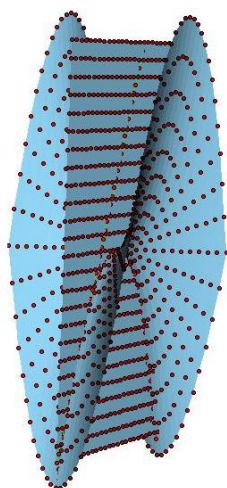
3.2.2 Formát JVX

Formát JVX, jak už jsme si řekli, je oficiální formát programu JavaView. Formát JVX použijeme, pokud chceme využít některé z nadstandardních možností, které přes Maple nejsou dostupné nebo odstranit některou z chyb, vzniklou při převodu do formátu MPL.

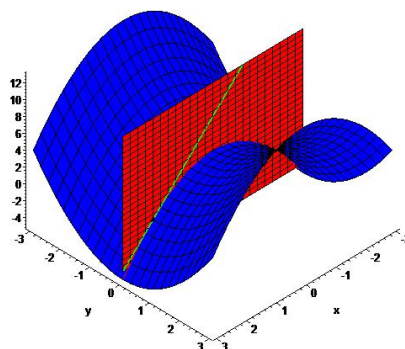
Formátu MPL jsme se věnovali v kapitole 3.2.1. Bohužel problém se zobrazováním vrstevnic přetrvává i při použití formátu JVX. Pracujeme-li s formátem JVX lze objekt měnit, upravovat a nastavovat parametry zobrazení přímo programem JavaView. Tato nastavení pak uložíme do souboru s příponou JVX a JVD. Při načítání appletů pak voláme oba soubory JVX a JVD spolu, toto načtení je podrobně popsáno v kapitole 5. Nyní si uvedeme základní typy pro nastavení vzhledu pomocí appletu JavaView. Nejprve vyexportujeme vhodný graf v Maple, na kterém budeme demonstrovat postupy pro změny vzhledu:

```
> Funkce:=plot3d(4-x^2+y^2, x=-3..3, y=-3..3, color=blue,
axes=framed):
> Bod:=pointplot3d([2,0,0], symbol=circle, color=black):
> Prunik:=spacecurve([t,0,4-t^2], t=-3..3, color=red):
> Rovina:=plot3d([x,0,z], x=-3..3, z=-4..12, color=red, axes=framed):
> Tecna:=spacecurve([t,0,-4*t+8], t=0..3, axes=NORMAL, thickness=2,
color=GREEN):
> Graf:=display(Funkce, Bod, Prunik, Tecna, Rovina):
> runJavaView(Graf, "Graf.jvx");
```

Výsledek exportu do JVX a jeho následném zobrazení v JavaView je na obrázku 22. Pro srovnání je také uvedeno vykreslení tohoto grafu v Maple.



Obrázek 22: Graf.jvx



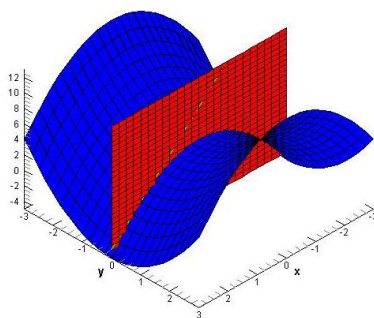
Obrázek 23: Zobrazení grafu Graf v Maple

Vidíme, že při exportu do formátu JVX se ztratily některé informace, jako například barva, styl vyjreslení apod. Dokonce graf se vykreslil pomocí bodů. Výhodou může být stejný poměr os, což jak už jsme si řekli, formát MPL nerespektuje. Vyexportujeme si tedy náš graf do formátu MPL, provedeme následující úpravy v JavaView a uložíme jej ve formátu JVX, který tyto úpravy zachovává. Export předchozího grafu provedeme příkazem:

```
> runJavaView(Graf, "Graf");
```

Výsledek exportu do MPL a následném zobrazení v JavaView je na obrázku 24.

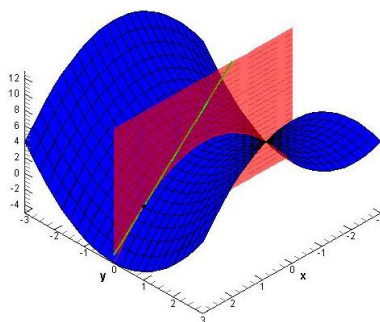
Cílem není vysvětlit všechny možnosti nastavení, je jich opravdu hodně, ale ukázat nejdůležitější z nich. Především popíši ty, které jsem využila při tvorbě mého webu.



Obrázek 24: Graf.mpl

Máme tedy graf zobrazen pomocí appletu JavaView a můžeme se pustit do úprav:

Nejprve si ukážeme nastavení průhlednosti roviny. V programu JavaView zvolíme **Inspector** → **Display** a musíme si nastavit v záložce **Geometries** aktivní geometrii, tedy v sloupečku Active Geometry zvolíme objekt, ve kterém chceme průhlednost nastavit. V našem případě zvolíme **Mesh** (rovina). Pro nastavení transparency přejdeme do **Inspector** → **Geometry** → **Material** a v části **Show** zaškrtneme **Transparency** a zrušíme volbu **Edge**. Míru průhlednosti volíme v části **Size and Length**, v položce **Transparency** hodnotou nebo posuvníkem. Nechala jsem zvolenou implicitní hodnotu 0,4. Výsledek vidíme na následujícím obrázku 25.

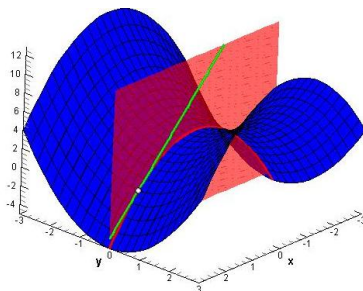


Obrázek 25: Nastavení Transparency

Části grafů se nám nevhodně překrývají. Chceme-li vyznačit křivku na grafu, grafika křivky a grafu se ruší a tím dojde k nevhodnému překrytí. Tyto potíže odstraníme tak, že v menu **Inspector** → **Display** a v položce **Flags** zaškrtneme **Enable 3D-Look, beta**.

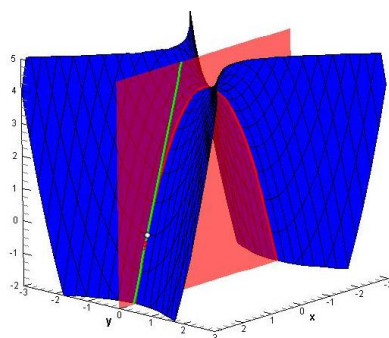
Chceme-li upravovat křivku, zvolíme si ji za aktivní v menu **Inspector** → **Display**, v záložce **Geometries** volbou **Polygon Set**. V položce **Inspector** → **Geometry** → **Material** můžeme nastavovat mimo jiné tloušťku nebo barvu křivky. Zvolíme **Polygon Size** na hodnotu 2, tím se stává křivka viditelnější.

Podobný postup použijeme na úpravu bodu. Zvolíme si ho za aktivní v menu **Inspector** → **Display**, v záložce **Geometries** volbou **PointSet**. V položce **Inspector** → **Geometry** → **Material** můžeme nastavovat mimo jiné velikost nebo barvu bodu. Velikost si zvětšíme v proměnné **Vertex Size** na hodnotu 4. Barvu z černé na bílou přenastavíme v položce **Global Color** v proměnné **Vertex**, buď kliknutím do rámečku se zobrazovanou barvou (zobrazí se paleta barev), nebo nastavením číselných hodnot na 255 255 255. Výsledek vidíme na následujícím obrázku 26.



Obrázek 26: Úprava tečny a bodu.

Bohužel nemáme zachovány poměry os, tento problém byl popsán také v kapitole 3.2.1. JavaView má nástroj jak tento zádrhel odstranit a vytvořit mnohem názornější obrázek. Nastavení provádíme v menu **Inspector** → **Camera**. Nejdříve si nastavíme zachování poměru os v **Box Ratio**, vypneme položku **Enable** a hodnoty **Ratios** nastavíme na 1 1 1. Je-li potřeba obrázek vycentrovat, učiníme tak pomocí **Center** a **Fit**. V sekci **Clipping** pak vhodně ořežeme obrázek, jelikož je zbytečně velký. Zaškrtneme položku **Scene**, zvolíme si osu Z a nastavíme Z-Max na 5 a Z-Min na -2. Výsledek máme ukázán na následujícím obrázku 27.



Obrázek 27: Nastavení os grafu.

Takto upravený graf si uložíme v menu **File** → **Save** → **JVX**. Pokud je JavaView spuštěn jako applet v HTML dokumentu, nelze ukládat přímo na disk. Exportovaný soubor je proto vypsán do textového okna a uživatel si jej může pomoci (CTRL+C/CTRL+V)

zkopírovat a uložit do textového editoru. Více o funkcích a nastavení JavaView si povíme v následující kapitole.

4 Java View

Pro správné zobrazení budoucího webového appletu je třeba mít nainstalováno tzv. běhové prostředí Javy. Některé systémy nebo novější prohlížeče je mají standardně zakomponovány. Na domovské stránce programu JavaView je zařazen Java applet, můžeme si tady ověřit, zda náš systém už běhové prostředí obsahuje. V pravém horním rohu by se měl objevit mnohostěn. V opačném případě si běhové prostředí opatříme na stránkách firmy Sun, <http://www.java.com>, jenž nese název Java Runtime Environment. Dále musí být v nastavení prohlížeče Java povolena, což by se mělo po instalaci provést automaticky. Běhové prostředí a webový prohlížeč je to jediné, co potřebují návštěvníci webových stránek s Java applety.

4.1 Ovládání appletů

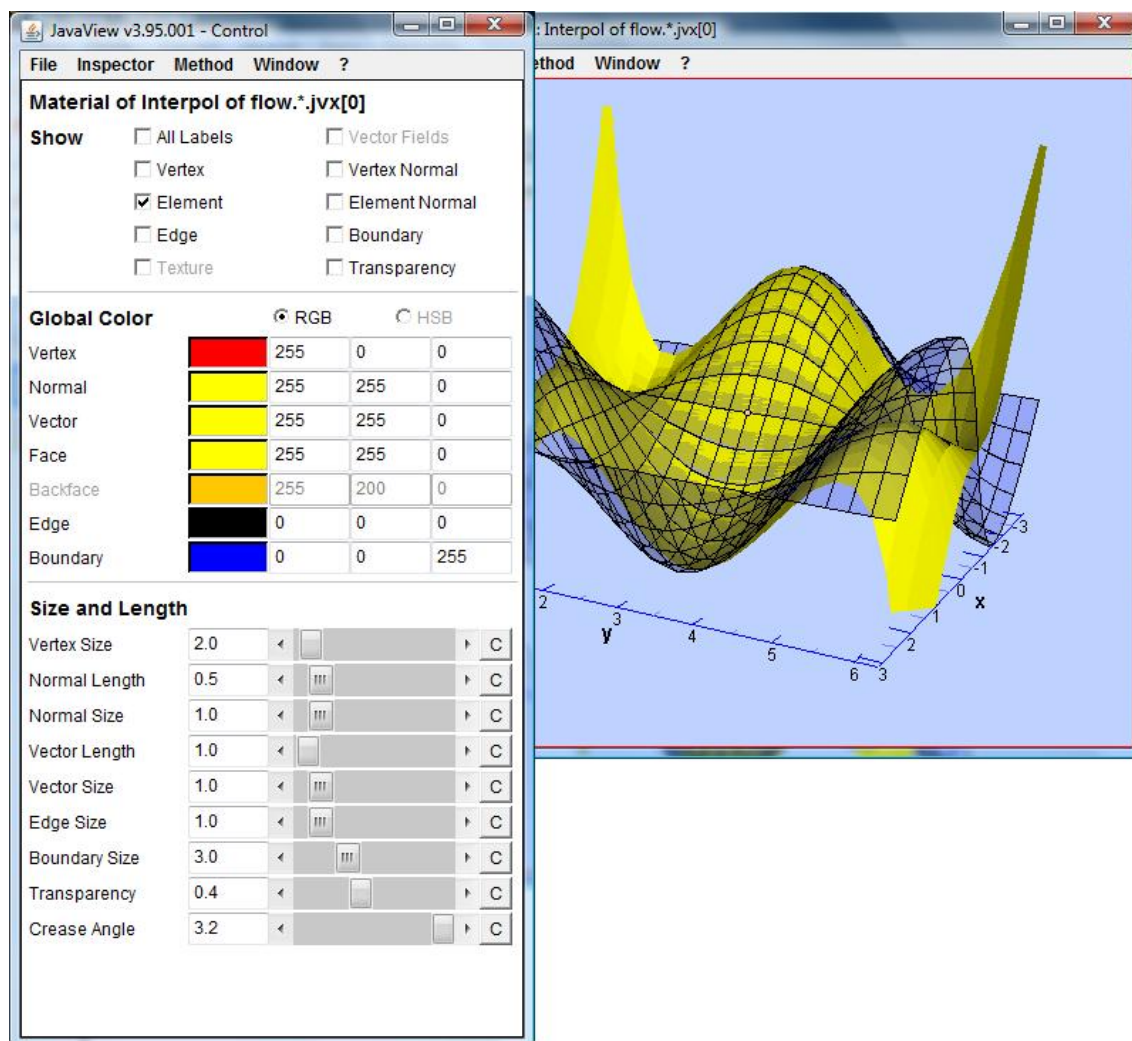
Applety můžeme jednoduše ovládat využitím klávesových zkratk. Pomocí nich se s 3D objektem jednodušeji manipuluje, především při prohlížení. Applet musí být aktivní (stačí kliknout do místa appletu levým tlačítkem myši). Pravým zobrazíme hlavní mód pro výběr akcí. Klávesovou zkratku musíme vždy držet po celou dobu a zároveň pravým tlačítkem myši applet ovládáme. V následující tabulce si jednotlivé zkratky popíšeme:

Zkratka	Popis	Zkratka	Popis
o	Rotace objektu	q	Ukončení animace
s	Změna velikosti	Shift+x	Zobrazení os
t	Posun objektu	Ctrl+tab	Přepínání mezi jednotlivými objekty
r	Obnovení výchozího nastavení	Ctrl + s	Zobrazení ovládacího panelu
a	Vložení dalšího vrcholu	Ctrl + d	Zobrazení JavaView konzoly
p	Uchopení vrcholu a jeho posun	Ctrl + a	Zobrazení panelu animace
i	Posun výchozího bodu	Ctrl + p	Zobrazení informací o projektu
m	Označení vrcholů v obdélníkovém výřezu	Ctrl + i	Informace o objektu
u	Vypnutí označení vrcholů v obdélníkovém výřezu	Ctrl + m	Zobrazí Materiál panel
c	Posun objektu do středu	Ctrl + c	Zobrazí Camera panel
f	Posun na střed a změna měřítka	Ctrl + v	Zobrazí Viewer panel
x	Zvětšení obdélníkového výřezu	w	Animace poslední události

Tabulka 6: Zkratky pro ovládání appletů

4.2 Control panel

Pohybujeme-li se v appletu a chceme měnit vzhled a další parametry, stiskneme klávesu F1, nebo pravým tlačítkem myši zvolíme Control panel. Pokud je JavaView spuštěn jako applet v HTML dokumentu, Control panel se nám občas nepříjemně ztrácí. Přejedeme-li kurzorem myši přes okno appletu, Control panel zmizí a musíme jej vyvolat znovu, nebo maximalizovat na dolní programové liště. Na následujícím obrázku 28 je zobrazen control panel:



Obrázek 28: Control panel.

Pro lepší ovládání a rychlejší orientaci, popíšu v následující tabulce 7 hlavní menu tohoto panelu. Popsala jsme pouze základní části, jelikož tento panel je podrobně popsán na adrese <http://www.javaview.de/doc/index.html>.

File		
New	Project	Nový projekt. Neuložená data jsou ztracena. Zavře všechna nadbytečná okna.
	Empty geometry	Nový objekt. Neuložená data jsou ztracena. Zavře všechna nadbytečná okna.
	Display AWT	Otevře další náhled na objekt Všechny změny s ním prováděné se zobrazují ve všech náhledech.
Add	Project	Přidá k původnímu objektu nový projekt. Aktivní projekt se nastavuje v Inspector - Display a ovládá se pomocí panelu Inspector - Project.
	Empty geometry	Přidá k původnímu objektu nový. Aktivní objekt se nastavuje v Inspector - Display.
Import	VGP Models	Otevře jeden z modelů z kolekce JavaView. Tento model je přidán k aktivnímu náhledu.
	Browse Disk	Dovoluje načíst z disku uložený objekt v podporovaných formátech. (Ne v HTML!)
Export	Dovoluje zobrazený objekt uložit v podporovaných formátech. Pokud je JavaView spuštěn jako applet v HTML dokumentu, nelze ukládat přímo na disk. Exportovaný soubor je proto vypsán do textového okna a uživatel si jej může pomocí (CTRL+C/CTRL+V) zkopírovat a uložit do textového editoru.	
Submit EG-Model	Otevře dialog dovolující snadnější přenos souborů přes EG-Model Server.	
Close	Uzavře ovládací panel.	
Exit	Ukončí program JavaView (Ne v HTML!)	
Inspector		
Project	Podává informace o aktivním projektu.	
Object	Info	Informace o vrcholech aktivního objektu, sousednosti a spojnicích.
	Material	Informace o vlastnostech jako barva ploch, síla spojnic a viditelnosti bodů aktivního objektu.
Camera	Nastavení náhledu a projekce aktivního objektu.	
Display	Nastavení aktivního objektu a ostatních zobrazovaných objektů.	
Method		
Mark	Mark/Unmark vertices	Označí/Odznačí vrcholy.
	Remove Marked Vertices	Odebere z objektu označené vrcholy.
	Mark/Unmark Elements	Označí/Odznačí plochy.
	Remove Marked Elements	Odebere z objektu označené plochy.
Window		
Animation	Ovládací panel animace.	
Console	JavaView konzole. Používá se při výpisu číselných údajů a při ladění projektů.	

Tabulka 7: Control Panel

5 HTML

HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. Jazyk HTML oproti jiným (XML, XHTML) nejlépe spolupracuje s Java skripty, s Java applety a skriptovacím jazykem PHP. Při tvorbě HTML stránek jsem využila také kaskádové styly CSS, které vyžadují alespoň částečnou znalost HTML. Mé stránky jsou rozděleny na 3 části podle následujícího obrázku:



Obrázek 29: Rozvržení HTML stránky

Rozčlenění provedeme efektivně pomocí PHP skriptu, který si popíšeme v následující kapitole.

5.1 PHP

Abychom na našem počítači mohli spouštět skripty v PHP, musíme PHP nejprve nainstalovat. Instalace PHP se skládá z knihoven PHP (zdrojová data), server (Apache) a případně databáze MySQL (nebudeme využívat). Celá instalace PHP může být provedena ručně, ale je to složitější proces, zvláště pro začátečníky. Druhou možností instalace PHP je využití programu PHPTriad, který je volně k stažení na adrese <http://www.slunecnice.cz/sw/php-triad/>.

PHP je programovací jazyk, který pracuje na straně serveru, kde máme uloženy zdrojové kódy webových stránek. PHP skript se nejprve spustí na serveru a po té výsledek odešle prohlížeči. Základní funkce jsou ukládání, mazání a změna dat.

5.1.1 Instalace a spuštění PHP skriptu

Program si stáhneme na již zmíněné adrese. Samotný program následně provede instalaci PHP a my jen odklikáme případná tlačítka "ok". Po skončení instalace PHP se objeví na disku nové složky (obvykle na disku C:\ - místním disku. Tyto složky obsahují důležitá data pro běh PHP. První z nich je PHPTriad, která obsahuje některé programy pro zálohování dat. Druhou složkou je složka Apache.

Chceme-li prohlížet PHP stránky, musíme provést tyto kroky:

- Spustit webový server Apache - to provedeme kliknutím na růžové peříčko ve složce Apache, server se spustí a můžeme jej minimalizovat do dolní programové lišty.

- Všechny stránky, které obsahují PHP skripty musíme překopírovat do složky `C:\Apache\htdocs\.` Do této složky budeme ukládat celý náš web.
- Webové stránky v PHP budeme nyní prohlížet trochu jinak - pokud si budeme chtít prohlédnout soubor `c:\apache\htdocs\index.php`, musíme si v prohlížeči naklikat adresu `http://localhost/index.php` a nemusíme být připojeni k internetu.

Často se na webových stránkách opakují některé části, hlavička, menu apod. Na rozdíl od jiných jazyků, s PHP si můžeme vytvořit šablonu pro web. Opakující se části uložíme pouze jednou a vkládáme příkazem `include`.

```
<body>
  <p>Hlavička na každé stránce stejná</p>
  <h1>menu</h1>

  <a href="include.php?stranka=uvod">Úvod</a>
  <a href="include.php?stranka=prvniStrana">První strana</a>
  <a href="include.php?stranka=druhaStrana">Druhá strana</a>

  <? $stranka = $_GET["stranka"];
  if ($stranka == "uvod") {include "uvod.txt";}
  else if ($stranka == "prvniStrana") {include "prvniStrana.txt";}
  else {include "druhaStrana.txt";}?>
</body>
```

Výpis 2: Ukázka členění webové stránky pomocí PHP.

Tato funkce uložená jako soubor `index.php` definuje menu (na každé stránce stejné) a vkládá obsah stránky, který máme uložen v souborech `uvod.txt`, `prvniStrana.txt` a `druhaStrana.txt`. Při prvním načtení stránky `index.php` se nám zobrazí s obsah souboru `druhaStrana.txt`. Zvolíme-li odkaz v menu na Úvod, vloží se obsah souboru `uvod.txt` a zvolíme-li odkaz na První stranu, vloží se obsah souboru `prvniStrana.txt`. V ostatních případech se vloží obsah souboru `druhaStrana.txt`, například není-li jeden ze souborů `uvod.txt` nebo `druhaStrana.txt` dostupný.

5.2 Začlenění interaktivní grafiky do HTML

V této kapitole si popíšeme, jak zobrazit objekty exportované z Maple na webových stránkách. Nejprve si ukážeme jednoduchý kód pro vložení našeho appletu na stránky:

```
<APPLET CODE='javaview.class' ARCHIVE='jars/javaview.jar,jars/jvx.jar' CODEBASE='.' WIDTH='
200' HEIGHT='200' ID='JVLAPPLET' ALT='JVL- MAPLE_Export'>
<PARAM NAME='Model' VALUE='objekt.mpl'>
</APPLET>
```

Výpis 3: Začlenění interaktivní grafiky do HTML.

Jak vidíme, celý applet je vložen pomocí párového HTML tagu `<APPLET>`. Dále si popíšeme jednotlivé atributy:

- CODE='javaview.class' - název třídy, která spouští applet je součástí knihovny JVL v adresáři jars tedy C:\Maple10\JavaViewLib\jars. Tento atribut neměnit !
- ARCHIVE='jars/javaview.jar;jars/jvx.jar' - jméno použité knihovny. Tento atribut neměnit !
- CODEBASE='.' - cesty ke všem souborům, které tento applet využívá.
- WIDTH='200' HEIGHT='200' - výška a šířka okna appletu, podle potřeby měníme hodnoty
- <PARAM NAME='Model' VALUE='objekt.mpl'> - odkazuje na soubor s 3D objektem.

Můžeme přidat další atributy, pomocí řádku

<PARAM NAME='Name' VALUE='Value'>, kde podle následujících tabulek měníme 'Name' a 'Value':

Name	Value	Popis
model	'objekt.jvx','objekt.mpl', ...	Načtení objektu z daného souboru v některém z podporovaných formátů souborů (MPL,JVX, OBJ,BYU ...)
displayFile	"objekt.jvd"	Načtení souboru s uloženými parametry zobrazení.
majorMode	rotate rotate-xy scale translate translate-z initial pick	V hlavním modu bude nastaveno: 3D rotace vertikální rotace změna velikosti posun ve směru xy posun ve směru z posun výchozího bodu uchopení bodu a jeho posun.
directSelect	Show [Hide]	Přímý výběr objektu kliknutím na ně.

Tabulka 8: Parametry appletu část 2.

Nastavení kamery		
Name	Value	Popis
interest	3 hodnoty	V tomto bodě bude cílové místo, kde je kamera zaměřena. Volí se vhodně, aby byl celý objekt viditelný(nejlépe jeho střed).
viewDir	3 hodnoty	Směr kamery, při perspektivním pohledu '0.137 0.824 -0.549'.
distance	hodnota	Vzdálenost kamery od (interest) cílového bodu.
autoRotate	Show [Hide]	Automatická rotace zobrazovaného objektu.
rotDir	3 hodnoty	Zadáváme osu rotace. Výchozí nastavení '0 1 0' - svislá osa.
Nastavení displeje		
Name	Value	Popis
antiAlias	Show [Hide]	Hladké hrany,pomalejší renderování.
axes	Show [Hide]	Zobrazení os.
background	3 hodnoty	Barva pozadí zadává se v RGB např. '255 0 0'
backgroundImage	URL nebo soubor	Obrázek na pozadí.
border	[Show] Hide	Hrana okolo appletu.
copyright	Show [Hide]	Zobrazení autorských práv.
depthcue	[Show] Hide	Zobrazí objekt dál od pozorovatele. Dojem hloubky.

Tabulka 9: Parametry appletu část 1.

Tyto parametry, jak už jsme si řekli, zapisujeme přímo do HTML kódu. Další možností je, že si je bude nastavovat uživatel na stránce sám. K tomu slouží Control panel, který se aktivuje pravým tlačítkem myši výběrem Control panel, který je popsán v kapitole 4.2.

5.3 Animace

Animace můžeme tvořit dvojím způsobem, buď přímo v Maple nebo až v JavaView. Na mých webových stránkách se přikláním k druhému způsobu, jelikož lze měnit parametry objektů, které nejsou přímo v Maple dostupné, jako například průhlednost, kamera apod.

Začneme ukázkou jednoduchého kódu:

```
<APPLET CODE='javaview.class' ARCHIVE='jars/javaview.jar' CODEBASE='.' WIDTH='400'
  HEIGHT='400'>
  <param name=model value="flow.*.jvx">
    <param name=Animation.firstKey value=1>
      <param name=Animation.lastKey value=3>
    </param>
  </applet>
```

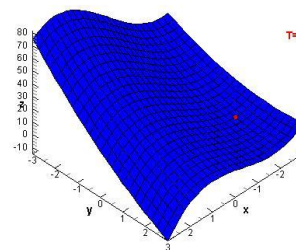
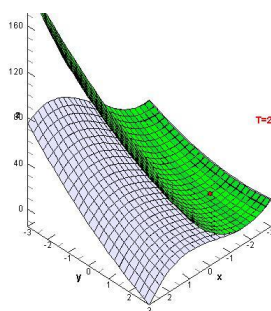
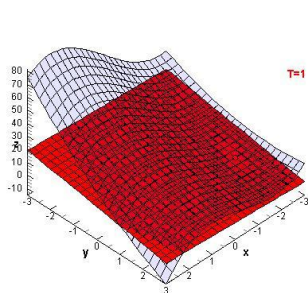
Výpis 4: Animace

Důležité je mít vytvořené objekty, které chceme animovat ve formátu JVX. V našem případě tedy máme 3 objekty pojmenované flow.1.jvx, flow.2.jvx, flow.3.jvx.

Podívejme se na 3. řádek kódu, první dva nemusíme komentovat, jsou stejné jako v předchozí kapitole.

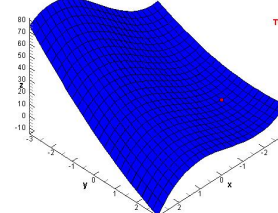
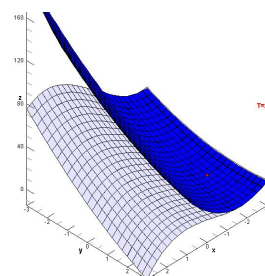
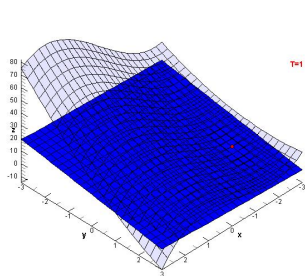
```
< param name=model value="flow.*.jvx">
```

Jak už jsme si řekli, jedná se o odkaz na soubor s 3D objektem. Animace se skládá z více snímků neboli souborů JVX s objektem, který mění své vlastnosti (barva, pozice, velikost, tvar ...). Tyto objekty se pak ukazují v závislosti na čase od flow.1.jvx po flow.3.jvx. Na následujících obrázcích si ukážeme jednotlivé snímky:



Obrázek 30: flow.1.jvx Obrázek 31: flow.2.jvx Obrázek 32: flow.3.jvx

Na obrázku 35 vidíme pouze jeden objekt narozdíl od obrázků 33 a 34, které se skládají z objektů dvou, ale není tomu tak, také obrázek 35 se skládá z objektů dvou, protože jsou totožné. Počet a názvy objektů se musí shodovat, jinak se animace neprovede. U animace JavaView nejde o přecházení ze snímku na snímek, ale z důvodu interaktivity JavaView dotváří snímky vlastní a zajišťuje, tak plynulý přechod. Bohužel u příkladů typu Taylorův Polynom nemá plynulá animace smysl a nelze implicitně nastavit její vypnutí. Vypnout si jej může uživatel sám. Zde jsem narazila na další nepříjemný nedostatek, a to že při vypnutí Linear Interpolation zároveň vypneme přechod mezi barvami. Předchozí snímky animace pak vypadají takto:



Obrázek 33: flow.1.jvx

Obrázek 34: flow.2.jvx

Obrázek 35: flow.3.jvx

Informaci, jak má javaView načítat jednotlivé soubory, máme uloženu v následujících atributech:

< param name=Animation.firstKey value=1> - Animace začne ve flow.1.jvx za * se dosadí hodnota uložená ve value. Pokračuje dosazením hodnoty o jednu vyšší tedy 2, až po poslední klíč, který zapisujeme zde:

< param name=Animation.lastKey value=3>.

Opět můžeme přidat další atributy viz tabulka 10.

Name	Value	Popis
Animation.fixedLen	4	Počet číslic pro číslo používané v keyframe a tedy i lastKey a firstKey. Například flow.0023.jvx má fixedLen = 4. Výchozí hodnota je 0, což není pevná délka.
Animation.dialog	[Show] Hide	Ukázat nebo skrýt animační dialog, výchozí je zobrazit.
Animation.speedType	[Forward] Rewind FastForward FastRewind	Směr a rychlost animace. Výchozí je vpřed.
Animation.repeatType	[BackForth] Loop OneWay	Opakování animace neboli její chování po dosažení posledního klíče.
Animation.framesPerSecond	20	Rychlost animace. Číslo udává počet zobrazených snímků za sekundu.
Animation.timeStepIncr	1	Přírůstek mezi dvěma po sobě jdoucími body. Výchozí je 1.
Animation.start	Show [[Hide]	Start animace hned. Výchozí je ne.

Tabulka 10: Parametry animace

5.4 JsMath

Potřebujeme-li vkládat rozsáhlé matematické vzorce na web, je dříve používané vkládání pomocí obrázků velice zdoluhavé a pracné. K tomuto účelu byly vyvinuty volně dostupné aplikace, jako například JsMath a MathML. Aplikace MathML má oproti JsMath jisté nedostatky, především nezobrazuje matematiku na všech prohlížečích správně a vyžaduje široké znalosti v programování.

Naproti tomu skript jsMath nabízí více způsobů zobrazení, každý má své opodstatnění a záleží pouze na volbě autora. Nejlépe jsMath pracuje s TeXovskými fonty, ale ty bohužel vyžadují instalaci uživatelem na svůj počítač, proto má jsMath v záloze možnost zobrazení pomocí obrázků nebo Unicode fontů. Nejjednodušší je zobrazení matematiky pomocí Unicode fontů, není sice tak úhledné jako fonty TeXovské, ale nevyžaduje aktivitu uživatele, nebo tisíce malých obrázků. Zobrazování pomocí Unicode totiž využívá základní sady písmen nainstalované ve Windows. Záleží také na prohlížeči, kterou sadu pro zobrazování textu využívá. Tuto sadu si můžeme rozšířit instalací dalších písmen například už zmíněných TeXovských fontů.

5.4.1 Instalace JsMath

Informace a samotnou aplikaci jsMath nalezneme na adrese: <http://www.math.union.edu/~dpvc/jsMath/>. Pro zobrazení matematiky na našem webu nám postačí balíčky "jsMath" a "jsMath Image Fonts" dostupné na adrese http://sourceforge.net/project/showfiles.php?group_id=172663. Pro instalaci postupujeme následovně:

- balíček jsMath si rozbalíme a uložíme kdekoli na našem webovém serveru
- balíček jsMath Image Fonts si rozbalíme a obsah uložíme do prvního balíčku jsMath a to do adresáře fonts tedy na místo `... \jsMath\fonts`

Po instalaci si spustíme testovou stránku `index.html` dostupnou v balíčku jsMath v adresáři `test`. Instalace proběhla úspěšně, zobrazí-li se na webové stránce matematický text a zelený nadpis "jsMath appears to be working!". Balíček jsMath Image Fonts slouží k zobrazování matematiky pomocí obrázků. Osahuje řadu obrázkových souborů v různých velikostech. Správnou instalaci tohoto balíčku si ověříme pomocí testové stránky `index-images.html` dostupnou v balíčku jsMath v adresáři `test`. Stejně jako u předchozího testu instalace proběhla úspěšně zobrazí-li se na webové stránce matematický text a zelený nadpis "jsMath appears to be working!".

5.4.2 Vkládání jsMath na HTML stránky

Pro vkládání matematiky na naše stránky pomocí jsMath si musíme nahrát skript do HTML kódu. Stačí zapsat do hlavičky HTML souboru následující příkaz:

`<SCRIPT SRC="jsMath/easy/load.js"> </SCRIPT>`. Po té již můžeme psát matematiku do těla kódu mezi párové tagy `<DIV>` a ``. Párový tag `<DIV>` slouží pro vkládání matematiky na samostatný řádek a tag `` mezi text. Pro vkládání používáme standardní TeXovské sazby.

Ukážeme si nyní ukázkou HTML kódu s jednoduchou matematikou:

```
<HTML>
  <HEAD>
    <SCRIPT SRC="jsMath/easy/load.js"> </SCRIPT>
  </HEAD>
  <BODY>
    Matematika pomocí jsMath: <DIV CLASS="math">  $f(x,y) = x^2 + y^2$  </DIV>
    A v textu <SPAN CLASS="math">  $f(x,y) = x^2 + y^2$  </SPAN> takto zapisujeme.
  </BODY>
</HTML>
```

Výpis 5: Sázení matematiky pomocí tagu DIV a SPAN.

Výsledek zobrazený v prohlížeči si ukážeme na následujícím obrázku:

Matematika pomocí jsMath:

$$f(x,y) = x^2 + y^2$$

A v textu $f(x,y) = x^2 + y^2$ takto zapisujeme.

Obrázek 36: jsMath pomocí tagu DIV a SPAN

JsMath má různé rozšíření a pluginy, bohužel jejich nastavení není pro úplného začátečníka lehké. Naštěstí toto nastavení za nás provede doplněk easy load, kde nastavíme hlavní detaily jsMath. Tento doplněk se nachází v balíčku jsMath v adresáři easy pod názvem load.js. Skript load.js obsahuje řadu funkcí, které máme již přednastavené a většinou upravujeme pomocí hodnot 0 (Vypnuto) a 1 (Zapnuto), nebo příslušnými výrazy. Skript upravíme pomocí WordPadu. V tabulce 11 máme popsány některé funkce.

Z těchto uvedených funkcí doporučuji přednastavit processSingleDollars z 0 na 1. Nemusíme pak používat párový tag SPAN na vkládání matematiky v textu a můžeme jej nahradit jednoduchým \$. Párový tag DIV zase nahradíme dvěma dolary \$\$\$. Uvedeme si ukázkou použití:

```
<HTML>
  <HEAD>
    <SCRIPT SRC="jsMath/easy/load.js"> </SCRIPT>
  </HEAD>
  <BODY>
    Matematika pomocí jsMath: $$$f(x,y) = x^2 + y^2$$$
    A v textu $f(x,y) = x^2 + y^2$ takto zapisujeme.
  </BODY>
</HTML>
```

Výpis 6: Sázení matematiky pomocí dolarů.

Výsledek zobrazený v prohlížeči bude stejný jako na obrázku 36.

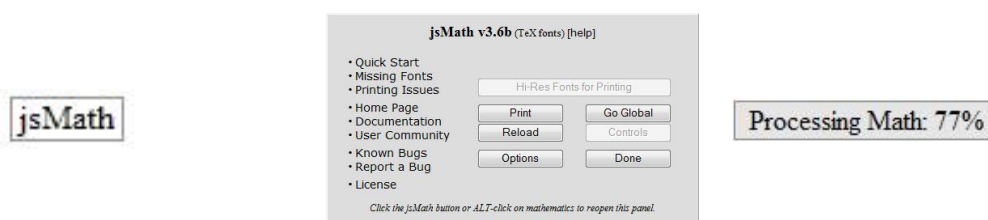
Název	Popis
root: ""	Nemáme-li umístěn kořenový adresář jsMath na stejném serveru jako load.js. Načítáme pomocí root:"http://umstn.jsMath".
scale: 120	Udává velikost matematiky k okolnímu textu. Měníme číselnou hodnotu podle potřeby.
autoload:1	Automaticky prohledá HTML kód a zjišťuje zda je potřeba překládat matematiku přes jsMath.
processSlashParens: 1 processSlashParens: 1 processDoubleDollars: 1 processSingleDollars: 0 processLaTeXenvironments: 0	Hledá v textu: $(...)$ $[...]$ $\$ \$ \dots \$ \$$ $\$ \dots \$$ $\backslash begin\{xxx\} \dots \backslash end\{xxx\}$ a překládá jako matematiku přes jsMath.
fixEscapedDollars: 0	vynechává \$ a nepřekládá přes jsMath
doubleDollarsAreInLine: 0	Překládá $\$ \$ \dots \$ \$$ matematiku v řádku.
allowDoubleClicks: 1	Po dvojklíku na matematiku se zobrazí původní nepřeložený zápis.
showFontWarnings: 1	Nemá-li uživatel nainstalovány TEXovské fonty zobrazí se cedulka s varováním.
method: "Process"	Nastavuje metodu zobrazení matematiky. Process - zobrazuje postupně při načítání a překládání stránky. ProcessBeforeShowing - zobrazí všechnu matematiku najednou až po překladu stránky.
loadFiles: []	Pro načtení dalších doplňků např.: loadFiles: ["plugins/mimeTeX.js"].
loadFonts: []	Pro načtení dalších fontů např.: loadFonts: ["cmmib10"].
allowGlobal: 1	Překládá matematiku do mezipaměti, tím se urychlí její zobrazení.
noImageFonts: 0	Vypíná načtení obrázků místo fontů.

Tabulka 11: Easy load

Dále jsem na svých stránkách vypnula zobrazení varovní `showFontWarnings: 0`, mimo jiné je možnost si toto varování upravit a umístit na stránky podle potřeby, což je podrobněji popsáno na stránkách jsMath. Na svých webových stránkách především v části menu je zbarvení textu a matematiky kromě klasické černé bílá. Vzhledem k tomu, že všechny obrázky jsou vykresleny pouze černou barvou a bylo by velmi zdoluhavé překreslení do barvy bílé. Vypnula jsem tedy načítání obrázků pomocí funkce `noImageFonts`: 1. Nemá-li uživatel nainstalovány nativní TeX fonty zobrazí se mu na svých webových stránkách nativní Unicode fonty pouze až po jejich instalaci se rozšíří nabídka o fonty TeXovské.

5.4.3 Úprava tlačítek jsMath

Ve výchozím nastavení jsMath se nám vytvoří malé tlačítko v pravém dolním rohu, které vždy zůstane na svém místě. Pomocí tohoto tlačítka si zobrazíme dialogové okno jsMath. Při načítání matematiky nebo při spouštění jiných operací v jsMath se spustí okno se zprávou. Máme tedy tři atributy, které se nám automaticky spustí na stránkách a slouží pro uživatele k ovládání jsMath. V této kapitole si ukážeme, jak si můžeme změnit jejich vzhled nebo umístění. Na následujícím obrázku si nejprve ukážeme výchozí vzhled těchto atributů:



Obrázek 37: jsMath-button Obrázek 38: jsMath-dialog Obrázek 39: jsMath-message

Vzhled a pozice těchto atributů měníme na základě kaskádových CSS stylů, o kterých jsme již mluvili výše. K tomu slouží příkaz:

```
<SCRIPT>jsMath={styles:{'#jsMath_button':
    {style-settings}}};</SCRIPT>
```

Příkaz se zapisuje do hlavičky HTML souboru těsně před příkaz pro nahrání jsMath. Vzhled a pozice se upravují, jak už jsme si řekli pomocí CSS stylů, nebo je můžeme jednoduše zapsat přímo do hlavičky. Ukážeme si tedy hlavičku HTML kódu a v ní atributy, jenž si můžeme nastavit:

```
<head>
<script type="text/javascript" src="ClickShowHideMenu.js"></script>
<SCRIPT>
jsMath = {styles: {
'#jsMath_button': { \\ nastaví vzhled tlačítka
    position: 'static', \\ pozice, při rolování vždy zůstane na svém místě ( absolute
, static, relative ...)
```

```

        bottom:    '1px',    \\ vzdálenost od spodního okraje (top)
        right :    '2px',    \\ umístění na pravo ( left )
        'background-color': 'blue',    \\ barva pozadí
        border:    'solid_2px_',    \\ tloušťka ohraničení
        'border-color': 'white_blue_blue_white',\\ barva ohraničení
        margin:    '0px_0px_0px_0px', \\ vnější okraj
        padding:    '0px_20px_0px_20px', \\ vnitřní okraje
        'z-index':  '102', \\ priorita při překrývání ( nemá vliv na java applety
        color:      'white',    \\ barva textu
        'text-decoration': 'none', \\ dekorace textu ( odkazy a pod.)
        'font-size': 'large',    \\ velikost textu ( small)
        width:      '100',    \\ šířka
        height:     '23px',    \\ výška
        cursor:     'hand'    \\ podoba cursoru (Auto, Crosshair ... )
    },
    '#jsMath_panel': { \\ nastaví vzhled dialogového okna
        /* stejné jako výše nebudeme popisovat*/
        position :    'fixed',
        bottom:       '20px',
        left :        '1.5em',
        padding:      '3em_1.6em',
        'background-color': 'green',
        border:       'outset_2px',
        'border-color': 'white_green_green_white',
        'z-index':    '103',
        color:        'black',
        'font-size':  '10pt',
        'font-style': 'normal',
        width:        '200'
    },
    '#jsMath_message': { \\ nastaví vzhled okna zpráv
        /* stejné jako výše nebudeme popisovat*/
        position :    'static',
        bottom:       '1px',
        left :        '212px',
        'background-color': 'red',
        border:       'solid_2px',
        'border-color': 'white_red_red_white',
        margin:      '0px_0px_0px_0px',
        padding:     '0px_100px_0px_100px',
        'z-index':    '102',
        color:      'white',
        'font-size':  'large',
        width:       '100%'
    }
    }
};
</SCRIPT>
<SCRIPT SRC="jsMath/easy/load.js"> </SCRIPT>
</head>

```

Výpis 7: Atributy jsMath

Změny se zadávají zápisem "klíč: hodnota" a oddělují čárkou. Můžeme také přidat více stylů CSS, to záleží na volbě autora.



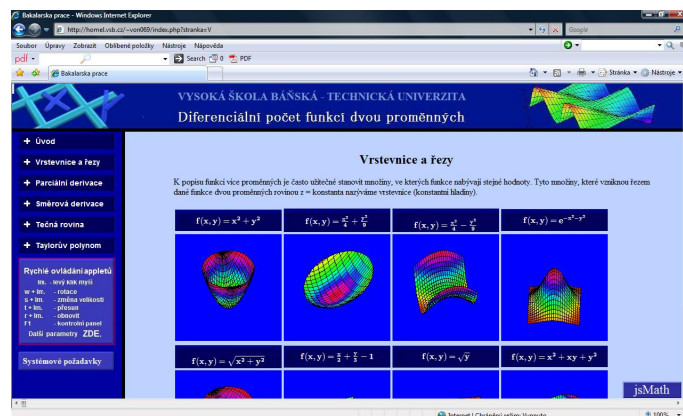
Obrázek 40: jsMath-button Obrázek 41: jsMath-dialog Obrázek 42: jsMath-message

Vzhledem k tomu, že nahrávání matematiky na stránkách pomocí jsMath je při velkém množství matematiky pomalé a nevhodně "naskakuje". Uvítala jsem možnost změny a nastavení atributů, aby uživateli bylo hned na první pohled jasné, co se na našich stránkách děje.

6 Ukázky příkladů

Způsobem popsáním v této práci vznikla webová galerie k tématu diferenciálního počtu funkcí dvou proměnných. Tato galerie je umístěna na přiloženém CD spolu s mapleovskými zápisníky použitými při její tvorbě. Galerie obsahuje 27 příkladů, rozdělených do pěti kapitol a ilustruje základní pojmy této části matematické analýzy. Každá kapitola má úvodní stranu se stručným vysvětlením probíraného pojmu a náhledem na interaktivní grafiku všech uvedených příkladů (spolu s odkazy). Tato úvodní strana má pomoci uživateli nejen k snadné orientaci, ale také seznámit jej s interaktivní grafikou, pomocí automatické rotace grafů.

Grafy jsou tak uvedeny automaticky do pohybu, hned po načtení stránky je na první pohled jasné, že se nejedná o obrázky statické ale dynamické. Na obrázku 43 je úvodní strana kapitoly Vrstevnice a řezy.



Obrázek 43: Náhled úvodní stránky kapitoly Vrstevnice a řezy

6.1 Rozvržení webové galerie

Vytvořená webová galerie je rozvržena do následujících kapitol:

1. Kapitola: Vrstevnice a řezy

Tato část objasňuje pojem vrstevnic. V řešených příkladech jsou vrstevnice spočítány a zobrazeny v rovině. Uvedeny jsou zde pro dokreslení představy i řezy významnými rovinami doplněny o dvojdimenzionální grafy křivek, které jsou výsledkem řezu.

2. Kapitola: Parciální derivace

Kapitola objasňuje pojem parciální derivace. Názorně vysvětluje na řešených příkladech geometrický význam tohoto pojmu. Uveden je zde mimo jiné výpočet směrnic tečen a jejich znázornění v prostoru i v rovině.

3. Kapitola: Směrové derivace

Kapitola vysvětluje pojem směrové derivace. Obdobně jako u kapitoly předchozí je zaměřena na vysvětlení geometrického významu tohoto pojmu. Třetí příklad názorně

demonstruje, že v případě směrové derivace se jedná o jisté zobecnění derivací parciálních. Čtvrtý příklad ukazuje souvislosti mezi pojmy gradient funkce a směrové derivace.

4. Kapitola: Tečná rovina

Tato část se věnuje pojmu tečná rovina. Význam tohoto pojmu je vysvětlen na čtyřech příkladech s interaktivní grafikou. Třetí příklad není pouze omezen na hledání rovnice tečné roviny, ale nejprve hledá průsečík přímky s funkcí, ve kterém se nachází výsledná tečná rovina. Čtvrtý příklad hledá nejen tečnou rovinu, ale zároveň rovinu rovnoběžnou s rovinou jinou (uvedenou v zadání).

5. Kapitola: Taylorův polynom

U prvního příkladu je spočítán Taylorův polynom 5. stupně. Graf funkce a její polynom jsou znázorněny pomocí animace, která zobrazuje i polynomy vyšších stupňů. Ve druhém příkladu je výpočet Taylorova polynomu 3. stupně a třetí příklad počítá Taylorův polynom stupně 2, pomocí něhož počítáme přibližnou hodnotu odmocniny bez použití kalkulačky. U všech příkladů je přiložena názorná animace.

Dále si uvedeme ukázky příkladů z vytvořených webových stránek.

6.2 Vrstevnice a řezy

Příklad 6.1

Určete vrstevnice a řezy funkce $f(x, y) = x^2 + y^2$ a znázorněte její graf.

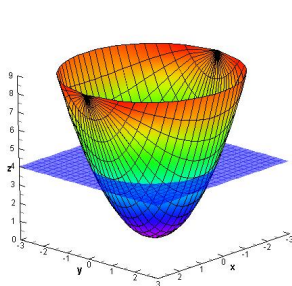
Řešení:

Definiční obor funkce je $D(f) = \mathbb{R}^2$.

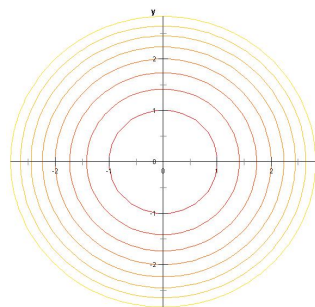
Vrstevnice v rovině xy :

Pro vrstevnice platí $z = c$ ($c = \text{konst.}$). Rovnice vrstevnic vypadá takto $c = x^2 + y^2$, kde c nabývá hodnot:

- $c < 0$ vrstevnice naší funkce je prázdná množina.
- $c = 0$ rovnici odpovídá pouze jeden bod a to $[0, 0]$.
- $c > 0$ jsou to kružnice se středem v počátku (osa z) a s poloměrem \sqrt{c} viz Obrázek 45.



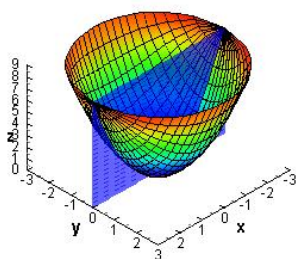
Obrázek 44: Řez rovinou $z = 4$.



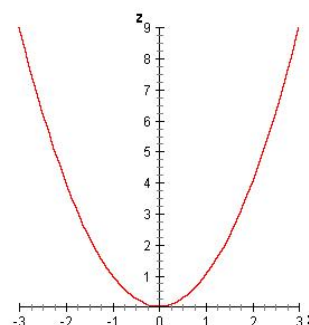
Obrázek 45: Znázornění vrstevnic v rovině xy .

Řez rovinou xz :

Dosadíme $y = 0$ a dostáváme funkci $z = x^2$, což je parabola s vrcholem v počátku viz Obrázek 47.



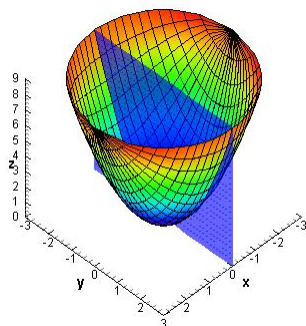
Obrázek 46: Řez rovinou $y = 0$



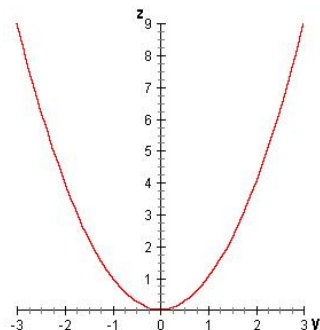
Obrázek 47: Znázornění řezu v rovině xz .

Řez rovinou yz :

Dosadíme za $x = 0$ a dostáváme funkci $z = y^2$, což je parabola s vrcholem v počátku viz Obrázek 49.



Obrázek 48: Řez rovinou $x = 0$.



Obrázek 49: Znáznornění řezu v rovině yz .

Na základě získaných výsledků již můžeme říci, že grafem naší funkce je **rotační paraboloid** s vrcholem v počátku a hlavní osou z , nacházející se v poloprostoru pro z kladné. ■

6.3 Parciální derivace

Příklad 6.2

Nalezněte parciální derivaci funkce $f(x, y) = x^2 + y^2$ podle x v bodě $A = (1, 1)$ a podle y v bodě $B = (1, -1)$.

Řešení:

Definiční obor funkce je $D(f) = \mathbb{R}^2$.

Parciální derivace funkce podle x :

Zadaný bod: $A = (1, 1)$. Parciální derivace podle x v bodě (x_0, y_0) je směrnice tečny ke křivce, která je průnikem roviny $y = y_0$

Ze zadání známe $x_0 = 1, y_0 = 1$. Vypočtěme si parciální derivaci funkce podle x :

$$\frac{\partial}{\partial x} f(x, y) = 2x,$$

po dosazení bodu A :

$$\frac{\partial}{\partial x} f(1, 1) = 2.$$

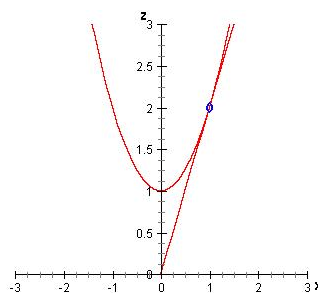
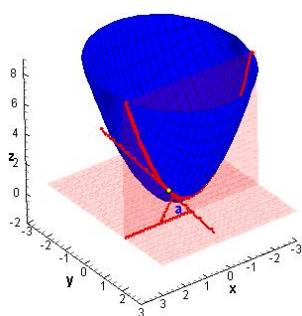
Ukažme si nyní, jaký je geometrický význam vypočtené hodnoty. Jak již bylo řečeno výše, jedná se o směrnici tečny ke křivce, která je průnikem roviny $y = 1$ a funkce $f(x, y)$ v bodě $A = (1, 1, 2)$.

Směrnice tečny je $\tan(\alpha)$, kde α je úhel, který svírá s kladnou částí osy x , viz Obrázek 50.

V našem případě tedy $\tan(\alpha) = 2$, tj. $\alpha \doteq 63^\circ$.

Podívejme se nyní na rovnici tečny. Obecně lze přímku v prostoru vyjádřit jako průsečnici dvou rovin nebo parametricky.

V prvním případě je tečna průsečnicí rovin $y = 1$ a $z = kx + q$. Přitom k je vypočtená směrnice a q dostaneme dosazením bodu A , tj. $2 = 2 \cdot 1 + q$. Druhou rovinou je tedy $z = 2x$.



Obrázek 50: Parciální derivace podle x v bodě A . Obrázek 51: Řez funkce rovinou $y = 1$ a tečna $z = 2x$.

Nyní uvedeme parametrické vyjádření přímky.

Parametrické vyjádření přímky je obecně dáno rovnicemi: $x = x_0 + u_1 t$, $y = y_0 + u_2 t$, $z = z_0 + u_3 t$,

kde (x_0, y_0, z_0) je libovolný bod přímky, (u_1, u_2, u_3) je směrový vektor přímky a $t \in \mathbb{R}$ parametr.

Dosadíme-li bod $(x_0, y_0, z_0) = (1, 1, 2)$ a směrový vektor $(u_1, u_2, u_3) = (1, 0, 2)$, dostáváme parametrické vyjádření tečny: $x = 1 + t$, $y = 1$, $z = 2 + 2t$.

Všimněte si, že v případě parciální derivace podle x jsou první dvě souřadnice směrovým vektorem osy x tj. $u_1 = 1$, $u_2 = 0$. Třetí souřadnice je hodnota parciální derivace podle x v daném bodě.

Parciální derivace funkce podle y :

Zadaný bod: $B = (1, -1)$

Vypočtěme si parciální derivaci funkce podle y :

$$\frac{\partial}{\partial y} f(x, y) = 2y,$$

po dosazení bodu A :

$$\frac{\partial}{\partial y} f(1, 1) = -2.$$

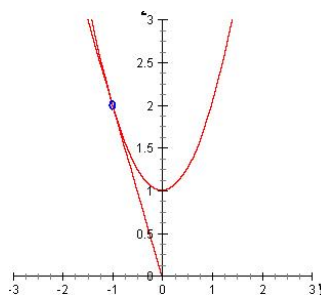
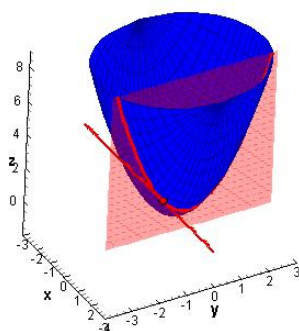
Ukažme si nyní, jaký je geometrický význam vypočtené hodnoty. Jak již bylo řečeno výše, jedná se o směrnici tečny ke křivce, která je průnikem roviny $x = 1$ a funkce $f(x, y)$ v bodě $A = (1, 1, 2)$.

Směrnice tečny je $\tan(\alpha)$, kde α je úhel, který svírá tečna s kladnou částí osy y , viz Obrázek 52.

V našem případě tedy $\tan(\alpha) = -2$, tj. $\alpha \doteq -63^\circ$.

Podívejme se nyní na rovnici tečny. Obecně lze přímku v prostoru vyjádřit jako průsečnici dvou rovin nebo parametricky.

V prvním případě je tečna průsečnicí rovin $x = 1$ a $z = ky + q$. Přitom k je vypočtená směrnice a q dostaneme dosazením bodu A , tj. $2 = -2 \cdot 1 + q$. Druhou rovinou je tedy $z = -2y + 4$.



Obrázek 52: Parciální derivace funkce po- Obrázek 53: Řez rovinou $x = 1$ a tečna
dle y v bodě B . $z = -2y$

Nyní uvedeme parametrické vyjádření přímky.

Parametrické vyjádření přímky je obecně dáno rovnicemi: $x = x_0 + u_1 t$, $y = y_0 + u_2 t$,
 $z = z_0 + u_3 t$,

kde (x_0, y_0, z_0) je libovolný bod přímky, (u_1, u_2, u_3) je směrový vektor přímky a $t \in \mathbb{R}$ parametr.

Dosadíme-li bod $(x_0, y_0, z_0) = (1, 1, 2)$ a směrový vektor $(u_1, u_2, u_3) = (0, 1, 2)$, dostáváme parametrické vyjádření tečny: $x = 1$, $y = 1 + t$, $z = 2 - 2t$.

Všimněte si, že v případě parciální derivace podle y jsou první dvě souřadnice směrovým vektorem osy y tj. $u_1 = 0$, $u_2 = 1$. Třetí souřadnice je hodnota parciální derivace podle y v daném bodě.

■

6.4 Směrové derivace

Příklad 6.3

Vypočtěte derivaci funkce $f(x, y) = x^2 + y^2$ v bodě $A = (1, 1)$ ve směru $u = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$.

Řešení:

Definiční obor funkce je $D(f) = \mathbb{R}^2$.

Určíme pomocnou funkci φ :

$$\varphi(t) = f\left(1 + \frac{1}{\sqrt{2}}t, 1 + \frac{1}{\sqrt{2}}t\right) = \left(1 + \frac{1}{\sqrt{2}}t\right)^2 + \left(1 + \frac{1}{\sqrt{2}}t\right)^2 = 2\left(1 + \frac{1}{\sqrt{2}}t\right)^2 = t^2 + 2\sqrt{2}t + 2$$

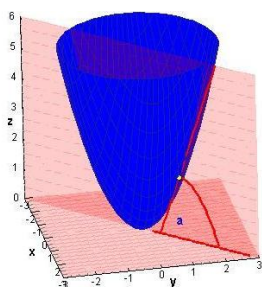
Spočteme derivaci pomocné funkce:

$$\varphi'(t) = 2t + 2\sqrt{2},$$

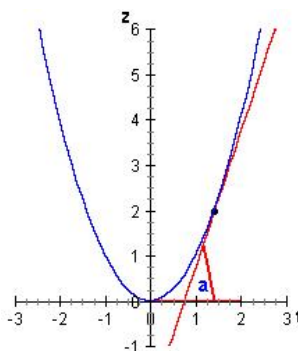
takže $\varphi'(0) = 2\sqrt{2}$. Hledaná derivace existuje a platí:

$$f_u(1, 1) = 2\sqrt{2},$$

což je směrnice tečny, tedy $\tan \alpha = 2\sqrt{2} \Rightarrow \alpha \doteq 70^\circ$. Přitom α je úhel, který svírá tečna s vektorem u viz Obrázek 54. Parametrické vyjádření tečny: $x = 1 + \frac{1}{\sqrt{2}}t$, $y = 1 + \frac{1}{\sqrt{2}}t$, $z = 2 + 2\sqrt{2}t$.



Obrázek 54: Směrová derivace podle vektoru u v bodě A .



Obrázek 55: Řez rovinou, rovnoběžnou s osou z a procházející přímkou $A + tu$, kde $t \in \mathbb{R}$.

■

6.5 Tečná rovina

Určete rovnice tečných rovin paraboloidu $f(x, y) = x^2 + y^2$ v průsečících s přímkou $p: x = t, y = t, z = t$.

Řešení:

Definiční obor funkce je $D(f) = \mathbb{R}^2$.

Obecná rovnice tečné roviny ke grafu funkce $f(x, y)$ v bodě (x_0, y_0, z_0) je:

$$z - z_0 = \frac{\partial}{\partial x} f(x_0, y_0)(x - x_0) + \frac{\partial}{\partial y} f(x_0, y_0)(y - y_0)$$

Vypočteme si nejprve průsečíky zadaného paraboloidu s přímkou, tak že dosadíme parametrické rovnice přímky do rovnice paraboloidu a dostaneme: $t = t^2 + t^2$, $t = 2t^2$, $0 = t(2t - 1)$.

$$t = 0 \text{ pro bod } (0, 0, 0) \text{ nebo } t = \frac{1}{2} \text{ pro bod } \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$$

Po dosazení parciálních derivací dostáváme rovnici tečné roviny:

$$z - z_0 = 2x_0(x - x_0) + 2y_0(y - y_0)$$

Konkrétně:

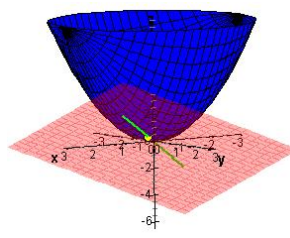
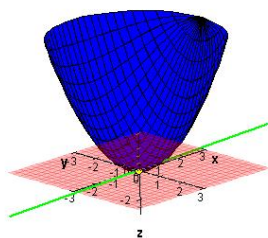
pro bod $(0, 0, 0)$

$$z = 0$$

pro bod $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$

$$z - \frac{1}{2} = 2 \cdot \frac{1}{2} \left(x - \frac{1}{2}\right) + 2 \cdot \frac{1}{2} \left(y - \frac{1}{2}\right)$$

$$\begin{aligned} z - \frac{1}{2} &= x - \frac{1}{2} + y - \frac{1}{2} = 0 \\ -2x - 2y + 2z + 1 &= 0 \end{aligned}$$



Obrázek 56: Funkce a tečná rovina v bodě $A = (0, 0)$.

Obrázek 57: Funkce a tečná rovina v bodě $A = \left(\frac{1}{2}, \frac{1}{2}\right)$.

■

6.6 Taylorův polynom

Příklad 6.4

Pomocí Taylorova mnohočlenu druhého řádu vypočítejte přibližně $\sqrt{3,05^2 + 2,95^2}$.

Řešení:

Definiční obor funkce je $D(f) = \mathbb{R}^2$.

Vyjdeme ze vztahu $f(x, y) \doteq T_2(x, y)$. Najdeme si vhodnou funkci $f(x, y) = \sqrt{x^2 + y^2}$. Zvolíme vhodně střed Taylorova polynomu $(x_0, y_0) = (3, 3)$ a dosazujeme bod $(h, k) = (0, 05; -0, 05)$. Nyní si spočteme jednotlivé derivace:

$\frac{\partial}{\partial x} f(x, y) = \frac{x}{\sqrt{x^2 + y^2}}$	$\frac{\partial}{\partial x} f(3, 3) = \frac{1}{2}\sqrt{2}$
$\frac{\partial}{\partial y} f(x, y) = \frac{y}{\sqrt{x^2 + y^2}}$	$\frac{\partial}{\partial y} f(3, 3) = \frac{1}{2}\sqrt{2}$
$\frac{\partial^2}{\partial x^2} f(x, y) = \frac{y^2}{\sqrt{(x^2 + y^2)^3}}$	$\frac{\partial^2}{\partial x^2} f(3, 3) = \frac{1}{12}\sqrt{2}$
$\frac{\partial^2}{\partial x \partial y} f(x, y) = \frac{-xy}{\sqrt{(x^2 + y^2)^3}}$	$\frac{\partial^2}{\partial x \partial y} f(3, 3) = -\frac{1}{2}\sqrt{2}$
$\frac{\partial^2}{\partial y^2} f(x, y) = \frac{x^2}{\sqrt{(x^2 + y^2)^3}}$	$\frac{\partial^2}{\partial y^2} f(3, 3) = \frac{1}{12}\sqrt{2}$

Tabulka 12: Parciální derivace

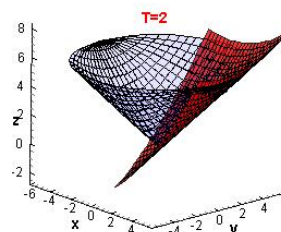
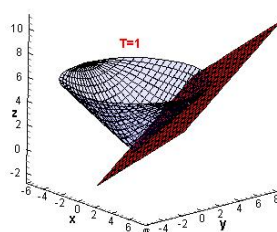
Tedy: $T_2(x, y) = f(3, 3) + \frac{1}{2}\sqrt{2}x + \frac{1}{2}\sqrt{2}y + \frac{1}{24}\sqrt{2}x^2 - \frac{1}{12}\sqrt{2}xy + \frac{1}{24}\sqrt{2}y^2$

Dosadíme do polynomu bod (h, k) a vypočteme přibližnou hodnotu:

$$T_2(0, 05; -0, 05) = 3\sqrt{2} + \frac{1}{2}\sqrt{2} \cdot 0,05 - \frac{1}{2}\sqrt{2} \cdot 0,05 + \frac{1}{24}\sqrt{2} \cdot 0,025 + \frac{1}{12}\sqrt{2} \cdot 0,025 + \frac{1}{24}\sqrt{2} \cdot 0,025$$

$$\sqrt{3,05^2 + 2,95^2} = 3\sqrt{2} + \frac{1}{6}\sqrt{2} \cdot 0,025 = \sqrt{2}\left(3 + \frac{1}{240}\right) = \sqrt{2}\frac{721}{240} = 4,23$$

Pro ověření si zkusme vypočítat zadaný příklad na kalkulačce: 4,243229901855425... Jistě rychlejší a snazší je výpočet na kalkulačce, která je dnes velmi snadno dostupná. Tento způsob výpočtu se používal dříve v době, kdy kalkulačky nebyly k dispozici.



Obrázek 58: Taylorův polynom 1. stupně. Obrázek 59: Taylorův polynom 2. stupně.

■

7 Závěr

Ve své práci jsem se zabývala vytvořením webové galerie za účelem prezentace problematiky diferenciálního počtu funkcí více proměnných. Vytvořila jsem ukázky příkladů s interaktivní 3D grafikou k ilustraci základních pojmů diferenciálního počtu jako jsou vrstevnice, parciální a směrové derivace, tečná rovina a Taylorův polynom. Tento web by měl přispět především studentům pro lepší pochopení probírané problematiky a geometrického významu základních pojmů.

Text bakalářské práce byl koncipován jako manuál, jenž shrnuje všechny kroky k vytvoření obdobných interaktivních stránek s podobným tématem, nebo jako samostatný manuál k jednotlivým programům a aplikacím.

Ve skutečnosti program Maple má spoustu možností využití a uplatnění, v mé práci se zaměřuji čistě na téma diferenciální počet funkcí více proměnných. Uváděla jsem především základní mapleovské příkazy k zobrazení 3D objektů (funkce, rovina, bod, křivka v prostoru) a různé parametry zobrazení.

Pro export interaktivní 3D grafiky z Maplu jsem použila mapleovskou knihovnu `JavaViewLib`. Tato knihovna poskytuje dva formáty výstupu (JVX a MPL), jejichž význam a přednosti jsem se snažila uvést na názorných příkladech. Začátek práce s knihovnou `JavaViewLib` není nijak snadný a bohužel autory může odradit, v knihovně je obsaženo mnoho funkcí, jejichž účinky se překrývají. Největší chybou poslední verze knihovny `JavaViewLib` (verze 3.22), která bohužel nešla nikterak obejít, považuji špatný převod mapleovských stylů, nejvíce však stylu `contour`, neboli vykreslení vrstevnic v prostoru.

Při exportu jsem narazila také na problémové vykreslení os, což pouze vyžadovalo speciální nastavení pomocí aplikace `JavaView`.

Tato aplikace umí zobrazovat exportované formáty (JVX, MPL) knihovny `JavaViewLib` a dokonce je můžeme pomocí ní efektivně upravovat. Pouze v případě formátu JVX máme možnost ukládat tyto úpravy grafiky. Bohužel v případě animací nastalo k problémovému běhu častěji. Dle mého názoru animace velice efektivně znázorňuje problematiku diferenciálního počtu funkcí více proměnných. K animacím můžeme přistupovat dvěma způsoby, buď pomocí programu Maple s omezenými možnostmi, nebo pomocí `JavaView`.

Já jsem zvolila způsob tvorby animací pomocí `JavaView`, kde bohužel jsou jisté části stále nefunkční. I tyto problémy jsem se pokusila co nejpodrobněji popsat a poukázat na možnou alternativu odstranění.

Dalším krokem tvorby webové galerie bylo vytvoření internetových stránek a začlenění interaktivní 3D grafiky. Popis slouží především k vytvoření jednoduchých HTML stránek pomocí skriptu PHP a stylu CSS, kterým jsem se věnovala nejméně. Na internetu je informací o těchto programech spousta i v českém jazyce, proto by bylo podrobnější popisování zbytečné a také to nebylo hlavní náplní mé práce.

Implementace latexovského zápisu matematiky do HTML kódu pomocí `jsMath` napomáhá k snadnému vytvoření stránek s matematickým textem. Mnou vytvořená webová galerie je rozšiřitelná o další kapitoly zabývajícími se lokálními extrémy, globálními extrémy a mimo jiné mohou být vytvořeny testy na probrané téma.

Stránky jsem se snažila vytvořit v co nejzajímavější podobě, aby uživatele zaujaly už při příchodu a neodradilo je pomalejší nahrávání matematiky a Java appletů. Při tvorbě jsem předpokládala co nejmenší aktivitu uživatele.

Text diplomové práce byl vysázen systémem . Jedná se o nadstavbu programu TeX, který umožňuje autorům textů sázet a tisknout dokumenty v nejvyšší možné typografické kvalitě a je vhodný zejména pro sazbu matematiky.

Michaela Vondrová

8 Reference

- [1] ŠARMANOVÁ, P., PLCH R.: *Interaktivní prezentace matematické grafiky na webu a v PDF dokumentech.*, Sborník semináře Technologie pro e-vzdělávání 2007, Praha: katedra počítačů ČVUT FEL, 2007, 31-38, ISBN 978-80-01-03756-0.
- [2] ŠARMANOVÁ, P., PLCH R.: *Maple, JavaViewLib a JavaView - nástroje k tvorbě, exportu a prezentaci interaktivní 3D grafiky.*, Sborník 6. matematického workshopu, 2007, 101-102, FAST VUT Brno, ISBN 80-214-2741-8.
- [3] ŠARMANOVÁ, P., PLCH R.: *Interaktivní 3D grafika v HTML a PDF dokumentech.*, Zpravodaj Československého sdružení uživatelů TEXu, Praha, Československé sdružení uživatelů TEXu. 2008, vol. 18, no. 1-2, s. 76-92, ISSN 1211-6661 (tištěná verze), ISSN 1213-8185 (online verze).
- [4] FILIPEC, Z., PLCH R.: *Maple a JavaView. Využití Maplu ve výuce a výzkumu na vysokých školách a akademiích věd.*, 1. vyd. Brno: Econ publishing s.r.o., 2002, ISBN 80-86433-10-2.
- [5] ŠELLEOVÁ, S.: *Prezentace matematické grafiky na webu s programem Java View.*, diplomová práce MU Brno, 2007.
- [6] *JavaView - Interactive 3D Geometry and Visualization*, duben 2009. Dostupné na [www : http://www.javaview.de/](http://www.javaview.de/).
- [7] *jsMath - A Method of Including Mathematics in Web Pages*, duben 2009. Dostupné na [www : http://www.math.union.edu/~dpvc/jsMath/](http://www.math.union.edu/~dpvc/jsMath/).
- [8] *Maple - Math Software for Engineers, Educators and Students*, duben 2009. Dostupné na [www : http://www.maplesoft.com/](http://www.maplesoft.com/).
- [9] *HTML - Jak psát web, návod na html stránky*, duben 2009. Dostupné na [www : http://www.jakpsatweb.cz/html/](http://www.jakpsatweb.cz/html/).